

Nagindas Khandwala College (Autonomous)

Affiliated to University of Mumbai



**MKES's Nagindas Khandwala College (Autonomous),
Gate No 5, Bhavishya Bharat Campus, S. V. Road, Malad (West)
Mumbai-400 064**

Programme Code: UBSCCS

Bachelor of Science in Computer Science (B.Sc.CS)

Three Year Integrated Programme -

Six Semesters

Course Structure

Under Choice Based Credit, Grading and Semester System

Implemented during Academic Year- 2022 -23

INDEX

Sr.No.	Content	Page No.
1	Preamble	2
2	Objectives	2
3	Program Outcome	2
4	Program Specific Outcome	3
5	Scheme of Examination	8
6	Course Structure	14
7	Detailed Curriculum	26

1. Preamble

There is growing demand of IT Professionals to deal with increasing software development, hardware and automation jobs. IT graduates with logical and problem solving skills are a need of the time. This program is so created to make our graduates are employable.

This Bachelor's Degree Program will provide students the right blend of knowledge and skills along with practical exposure.

2. Objectives

This program of B Sc CS is structured to provide graduates with practical skills required in software development, hardware development and testing field. The main objective of B Sc CS are:-

- To provide intensive theoretical & practical knowledge
- To provide an integrated perspective of IT functioning along with a good amount of exposure to real life cases / technical knowhow.
- To train students with knowledge in the field of logical thinking, problem solving, hardware designing, software engineering and programming.
- To develop students' programming skills to become employable or to start their own entrepreneurial journey.

3. Program Outcome

After completion of three years Bachelor of Science Computer Science (BSC CS) Programme, the learner will:

PO-1: Demonstrate a fundamental and systematic or coherent understanding of the academic field of Information Technology and its linkages with related disciplinary areas/subjects;

PO-2: Procedural knowledge that creates different types of professionals related to the disciplinary/subject area of Information Technology including software development and testing;

PO-3: Skills in areas related to one's specialisation within the disciplinary/subject area of Information Technology and current and emerging developments in the field.

PO-4: Demonstrate the ability to use the knowledge in formulating and tackling IT related problems and suggest software solution to them.

PO-5: Analyse and interpret data/information collected using appropriate methods, including the use of appropriate software such as programming languages and special-purpose packages, and report accurately the findings of the experiment/field investigations while relating the conclusions/findings to relevant theories.

PO-6: Demonstrate relevant global competencies such as problem solving skills that are required to solve different types of problems with well-defined solutions;

PO-7: Develop communication skills involving the ability to listen carefully, to read texts and research papers analytically and to present complex information in a concise manner to different groups/audiences;

PO-8: Develop (i) ICT skills such as presentation skills, documentation, etc; (ii) ^[11]_{SEP} personal skills such as the ability to work both independently and in a group (iii) skills to manage IT infrastructure.

PO-9: Demonstrate professional behaviour such as (i) being objective, unbiased and truthful in all aspects of work and avoiding unethical behavior such as fabricating, falsifying or misrepresenting data or to committing plagiarism; (ii) the ability to identify the potential ethical issues in work-related situations;

PO-10: Inculcate the mentality for (i) the appreciation of intellectual property, environmental and sustainability issues; and (ii) promoting safe learning and working environment.

4. Program Specific Outcome

After completion of this program learners will be:

1. Enhanced with scientific thinking and problem solving skills
2. Able to design computer programs to solve real life problems
3. Capable of designing secure organizational IT structure
4. Capable of handling core Computer Science concepts

P0-C0 Mapping

Semester	Subject	Course Code	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	
Sem 1	Communication Skills	2111UISCS	*			*			*	*	*	*	
	Web Programming I	2112UISWP	*		*	*	*		*	*		*	
	Introduction to Programming	2113UISIP	*	*		*	*	*			*	*	
	Fundamentals of Computers and Electronics	2114UISCE	*	*	*	*		*					
	Operating Systems	2115UISOS	*	*	*	*			*	*		*	
	Discrete Mathematics-I	2116UISDM	*	*					*				
	Practical of CC-1 + CC-2 Introduction to Programming + Fundamentals of Computers and Electronics	2117UISIPCE	*	*	*	*	*	*	*			*	*
	Practical of CC-2 + SEC-1 Operating Systems + Web Programming - I	2118UISOSWP	*	*	*	*			*	*	*		*
Sem 2	IT platforms, Tools and Practices	2121UISTP	*	*		*	*		*	*	*	*	
	Web Programming II	2122UISWP	*				*			*	*	*	
	Programming and Application Development in Python	2123UISDP	*	*	*		*		*	*		*	
	Object Oriented Programming	2124UISOP	*	*		*			*	*	*	*	
	Database Management Systems I	2125UISDM	*		*	*			*	*		*	
	Discrete Mathematics II	2126UISDM	*	*					*	*			
	Practical of CC-4 + CC-5 Programming and Application Development in Python + Object Oriented Programming	2127UISDPOP	*	*	*	*	*		*	*	*	*	

	Practical of CC-6 + SEC-2 Database Management Systems - I + Web Programming - II	2128UISD MWP	*	*	*		*		*	*	*	*	
Sem 3	Core Java	2131UISCJ	*	*	*					*	*	*	
	Data Structures	2132UISD S	*	*	*		*			*		*	
	Computer Networks	2133UISC N	*	*	*				*	*		*	
	Database Management Systems- II	2134UISD B	*	*	*					*	*	*	
	Descriptive Statistics	2135UISST	*	*	*	*	*	*					
	Advanced Web Programming - I	2136UISA W	*	*	*					*		*	
	Hybrid Mobile Application Development 1	2136UISM D	*	*	*	*				*	*	*	
	Computer Graphics and Animation	2137UISC G	*	*	*	*	*	*		*		*	
	Embedded Systems	2137UISES	*	*	*					*		*	
	Theory of Computation I	2137UCST C	*			*				*	*	*	*
	Core Java Practical	2131UISPR	*	*	*					*	*	*	
	Data Structures Practical	2132UISPR	*	*	*		*			*		*	
	Computer Networks Practical	2133UISPR	*	*	*				*	*		*	
	Database Management Systems- II Practical	2134UISPR	*	*	*					*	*	*	
	Advanced Web Programming - I Practical	2136UISA WPR	*	*	*					*		*	
	Hybrid Mobile Application Development 1 Practical	2136UISM DPR	*	*	*	*				*		*	
Computer Graphics and Animation Practical	2137UISC GPR	*	*	*	*	*	*		*		*		
Embedded Systems Practical	2137UISES PR	*	*	*					*		*		

	Theory of Computation I	2137UCST CPR				*			*	*	*	*
Sem 4	Design Patterns	2141UISDP	*	*	*	*		*		*		
	Design and Analysis of Algorithms	2142UISDA	*	*	*	*	*	*		*	*	*
	System Programming	2143UISSP	*	*	*	*				*		*
	Software Engineering	2144UISSE	*	*	*	*	*			*		*
	Probability Theory	2145UISPT	*	*	*	*	*	*		*		*
	Advanced Web Programming - II	2146UISAW	*	*	*	*						*
	Hybrid Mobile Application Development II	2146UISMD	*	*	*	*				*	*	*
	Advanced Java	2146UISAJ	*	*	*					*		*
	Multimedia Systems	2147UISM M	*	*	*	*			*	*		*
	Internet of Things	2147UISIT	*	*	*	*				*		*
	Theory of Computation II	2147UCST C	*		*	*			*	*		*
	Design Patterns Practical	2141UISPR	*	*	*	*		*		*		
	Design and Analysis of Algorithms Practical	2142UISPR	*	*	*	*	*	*		*	*	*
	System Programming Practical	2143UISPR	*	*	*	*				*		*
	Software Engineering Practical	2144UISPR	*	*	*	*	*			*		*
	Advanced Web Programming - II Practical	2146UISAWPR	*	*	*	*						*
	Hybrid Mobile Application Development II Practical	2146UISMDPR	*	*	*	*				*	*	*
	Advanced Java Practical	2146UISAJPR	*	*	*					*		*
	Multimedia Systems Practical	2147UISM SPR	*	*	*	*			*	*		*
	Internet of Things Practical	2147UISITPR	*	*	*	*				*		*

	Theory of Computation II Practical	2147UCST CPR	*		*	*			*	*		*	
Sem 5	Data and Network Security	2151UCSD N	*	*	*	*				*	*	*	
	Optimization Techniques	2152UCSO T	*	*	*	*	*	*		*	*	*	
	Web Services	2153UCS WS	*	*	*	*				*	*	*	
	Ethical Hacking	2153UCSE H	*	*	*	*	*	*	*	*	*	*	
	Introduction to Big Data Analytics	2153UCSB D	*	*	*	*				*	*	*	
	Wireless Sensor Networks	2154UCS WN	*	*	*	*				*	*	*	
	Game Programming	2154UCSG P	*	*	*	*	*	*		*	*	*	
	Web Technologies	2154UCS WT	*	*	*	*				*	*	*	
	Data and Network Security Practical	2151UCSP R	*	*	*	*				*	*	*	
	Optimization Techniques Practical	2152UCSP R	*	*	*	*	*	*		*	*	*	
	Web Services Practical	2153UCS WSPR	*	*	*	*				*	*	*	
	Ethical Hacking Practical	2153UCSE HPR	*	*	*	*	*	*	*	*	*	*	
	Introduction to Big Data Analytics Practical	2153UCSB DPR	*	*	*	*				*	*	*	
	Wireless Sensor Networks Practical	2154UCS WNPR	*	*	*	*				*	*	*	
	Game Programming Practical	2154UCSG PPR	*	*	*	*	*	*		*	*	*	
	Web Technologies Practical	2154UCS WTPR	*	*	*	*				*	*	*	
	Project Implementation	2155UCSPI	*	*	*	*	*	*	*	*	*	*	*
	Building blocks of Artificial Intelligence	2161UCSB A	*	*	*	*				*	*	*	

Sem 6	Data Science	2162UCSD C	*	*	*	*	*	*		*	*	*
	Cloud Computing	2163UCSC C	*	*	*	*				*	*	*
	Cyber Forensics	2163UCSC F	*	*	*	*	*	*	*	*	*	*
	Mobile and Pervasive Computing	2164UCSM P	*	*	*	*	*	*		*	*	*
	Digital Image Processing	2164UCSD I	*	*	*	*	*	*		*	*	*
	Building blocks of Artificial Intelligence Practical	2161UCSP R	*	*	*	*				*	*	*
	Data Science Practical	2162UCSP R	*	*	*	*	*	*		*	*	*
	Cloud Computing Practical	2163UCSC CPR	*	*	*	*				*	*	*
	Cyber Forensics Practical	2163UCSC FPR	*	*	*	*			*	*	*	*
	Mobile and Pervasive Computing Practical	2164UCSM PPR	*	*	*	*	*	*		*	*	*
	Digital Image Processing Practical	2164UCSD IPR	*	*	*	*	*	*		*	*	*
	Project Implementation	2165UCS PI	*	*	*	*	*	*	*	*	S*	*

5. Scheme of Examination

The scheme of Examination shall be divided into parts i.e. Continuous Internal Evaluation including Assignment, Projects, Seminars, Case Studies and Class Tests, Semester End Examinations and Practical Examination for some courses which will be of 50 marks. The semester wise Credit Points will be varied from course to course but the value of Credits for Under-Graduate Programme shall be of

(1) 120 Credits. Students will have to earn 8 extra credits under autonomy. This will be achieved by doing 1 Course on Environmental Science, 1 Short Term Course from an approved list of certifications and 4 IIT Online Spoken Tutorials. (Students admitted in First Year before 2021-2022)

(2) 132 credits for Students admitted in First Year before 2021-2022

The Credits are defined in terms of the learner's hours which are divided into two parts such as Actual and Notional. The value of a particular course can be measured in number of Credit Points. The value of One (01) Credit is equal to 15 Hours of learners' load

Scheme of Total Credits (Students admitted in First Year before 2021-2022)

Sr. No.	Year	Credits
1	Year 1	40
2	Year 2	40
3	Year 3	40
	Total Credits from Academics	120
	Additional Credits	8
	Total Credits for Award of Degree	128

5.1 Credit Based Evaluation System Scheme of Examination

For all 6 semesters, the performance of the learners shall be evaluated into two components. For Semester 1 to 4: the first component shall carry 40% marks which will be a Continuous Internal Evaluation while the second component shall carry 60% marks at semester end examination

For Semesters 5 and 6: The first component shall carry 25% marks which will be a Continuous Internal Evaluation while the second component shall carry 75% marks at semester end examination.

The allocation of marks for the Continuous Internal Evaluation 25% and Semester End Examination 75% and Practical Examination are as shown below:

5.2 Structure of Continuous Internal Evaluation

Sr. No.	Particulars	Marks
1	Class test held in the given semester and online examinations	20 marks
2	Subject specific Term Work Module/assessment modes –as decided by the department in the beginning of the semester (like Extension/field/experimental work, Short Quiz; Objective test, lab practical, open book test etc. and written assignments, Case study, Projects, Posters and exhibits etc. for which the assessment is to be based on class presentations wherever applicable)	15 marks
3	Active participation in routine class instructional deliveries (and in practical work, tutorial, field work etc. as the case may be)	5 marks
	Total:	40 marks

Sr. No.	Particulars	Marks
1	Class test held in the given semester, OR	10 marks
2	Subject specific Term Work Module/assessment modes –as decided by the department in the beginning of the semester (like Extension/field/experimental work, Short Quiz; Objective test, lab practical, open book test etc. and written assignments, Case study, Projects, Posters and exhibits etc. for which the assessment is to be based on class presentations wherever applicable)	10 marks
3	Active participation in routine class instructional deliveries (and in practical work, tutorial, field work etc. as the case may be)	5 marks
	Total:	25 marks

5.2.1 Structure of Practical Examination = 50 marks

Sr. No.	Particulars	Marks
1	Implementation Based	40
2	Viva Voce	05
3	Journal	05

Semester End Examination will be organized after all modules of the course are taught in the class. It will be a written examination / or as per the needs of the course a practical examination or a combination of both. This examination will be for 75 marks, for Semesters V and VI and for 60 marks for Semesters I to IV.

The assessment of Continuous Internal Evaluation and Semester End Examination as mentioned above for the Semesters I to VI shall be processed by the College – ‘Institutions of their Learners’ and issue the grade cards to them after the conversion of marks into grade as the procedure mentioned below.

The learners to pass a course shall have to obtain a minimum of 40% marks in aggregate for each course where the course consists of Continuous Internal Evaluation & Semester End Examination. The learner shall obtain minimum of 40% marks (i.e. 10 out of 25) in the Continuous Internal Evaluation, 40% marks in Semester End Examination (i.e. 30 Out of 75) 40% marks in Practical Examination (i.e. 20 out of 50) separately, to pass the course and minimum of Grade D in each program, wherever applicable, to pass a particular semester. A learner will be said to have passed the course if the learner passes the Continuous Internal Evaluation & Semester End Examination together.

5.3 Passing Standards

Grade	Marks (CIE + SEE)	Practical Marks	Grade Points
O	80 & Above	40 & Above	10
A+	70 to 79.99	35 to 39	9
A	60 to 69.99	30 to 34	8
B+	55 to 59.99	28 to 29	7
B	50 to 54.99	25 to 27	6
C	45 to 49.99	23 to 24	5
D	40 to 44.99	20 to 22	4
F	Less than 40	Less than 20	0

5.4 Approved Certifications for Additional Credits (Students admitted in First Year before 2021-2022)

Table for Additional Autonomy Credits

Table for Autonomy Credits					
Semester	Component	Duration	No. of Courses	Credit/ Course	Total Credits
<i>A. Ability Enhancement Compulsory Course (AECC)</i>					
3	AECC I Environment Science	1 Sem	1	2	2
2-5	AECC II Online Spoken Tutorials	4 Sem	4	1	4
<i>B. Skill Enhancement Course (SEC)</i>					
1-5	SEC I (Short Term Courses)	30 hrs (Table – 1)	--	2	2
				Total	8

All B. Sc. CS students graduating from Nagindas Khandwala College will earn 8 (Eight) additional credits under autonomy. It is compulsory for every student to earn these credits for grant of a degree. The scheme of credits is explained below.

Course on Environmental Science (EVS)

This course is offered in sem. III for 2 credits. It is compulsory for all students to complete this course. Lectures for this are arranged in college.

IIT Spoken Tutorial Programme

This is a compulsory course offered to all students of B.Sc IT and CS from Semester 2 to Semester 5. Students will choose the suitable course from the list of Spoken Tutorial courses offered by IITB. Students can download the lectures and learn at their pace. An online examination will be conducted by IITB and students will be appearing for the examination at the Computer Lab of the College under the invigilation of a faculty in-charge.

Short Term Courses (STC)

List of Short Term Courses Offered Semester Wise

Sr. No.	STC offered in Semester 1,3 & 5	STC offered in Semester 2,4 & 6
1	Basic Course in Brahnavidya	Add On Certificate course on Child Care & Child Development
2	Pranic Healing	Add On Certificate course on Travel & Tourism Management
3	Understanding Mental Health	Add On Certificate course on Computer Application
4	Leadership insights	Add On Certificate course on Functional English
5	Certificate in QGIS	Certificate Course in Gandhian Studies
6	Finding an 'Entrepreneur' in you	Certificate in GIS
7	Working with Vectors (COREL DRAW)	Tie and Dye
8	Certificate in Visual Merchandising	Jewellery Making
9	Certificate Course in Indian Embroidery	Working with Bitmap (Photoshop)
10	Certificate in Stylization	Grooming and Personality Development
11	CASI New York Global Student Associate Certification	Mind your Money
12	Certificate in Photography	Certificate Course in Banking
13	CASI New York Street Volunteering CASI	Understanding Pollution through Mobile Application
14	Certificate in Spanish Language	Digital Marketing
15	Fundamental of Capital Market and Derivatives	Certificate in Mandarin

16	Technical Analysis and trading on Stock Markets	Group Discussion Strategies
17	Financial Statement Analysis	Modelling
18	Certificate in Finance CASI	Certificate course on music
19	Certificate in Methods of Quantitative techniques	Certificate Course in Sports Management
20	Mastering Aptitude Skills	Mastering Analytical Skills
21	Enhancing Communication Skills	Certificate Course in Mobile Journalism
22	Gujrati Folk Dance	Certificate in Treasury Management CASI
23	Draping	CASI New York Retail Banking
24	Programme for Life Personal & Professional Insurance	Investment Banking Operations
25	Branding Yourself	Certificate in IFRS
26	COIST (Certificate Course In Online Sources , Tools and Techniques)	Making Effective PPT
27	Equity Share Boot Camp	Certificate in Beauty Treatments
28	Certificate in Hair Styling	

Khandwala College offers 55 Short Term Courses. These courses are divided in even and odd semesters and are organized in vacations. Students have to complete any **ONE** Short Term Course in any one of the semesters between 1 to 5. Courses offered in Sem. 1,3 & 5 are organized during the Diwali break and courses offered in Sem. 2 & 4 are organized in the month of April.

5.5 Exemptions from STC

Students satisfying any one of the criteria listed below will be granted exemption from registering for a short-term course. 2 credits under autonomy will be granted to students who have:-

1. Registered with NSS unit of Khandwala College, who have completed 120 hours with participation at the rural camp, and authorities in charge of NSS at college level have recommended their name for exemption.
2. Won prizes at University, State, National or International level in any sports activity- individual or team - and their name has been recommended by authorities in charge of Gymkhana at college level.
3. Enrolled for NCC unit of Khandwala College, have completed mandatory hours of training and have attended all programs of NCC at Khandwala College and their name has been recommended by authorities in charge of NCC at college level.
4. Represented Khandwala College at any University Cultural Festival and have won prize will be entitled to exemption on recommendation of their name by the Students Council in charge.

Bachelor of Science (Computer Science) Programme

Under Choice Based Credit, Grading and Semester System

Three Year Integrated Programme -

Six Semesters

Basic Structure: Distribution of Courses

1	Ability Enhancement Compulsory Course (AECC)	2 Paper of 2 Credits Hrs. each (Total Credits Hrs. $2*2=4$)	04
2	Skill Enhancement Compulsory Course (SEC)	2 Papers of 3 Credits Hrs. each (Total Credits Hrs. $2*3=6$) 4 Papers of 2 Credits Hrs. each (Total Credits Hrs. $4*2=8$) 8 Papers of 1 Credits Hrs. each (Total Credits Hrs. $8*1=8$)	22
3	Core Course (CC)	6 Papers of 3 Credits Hrs. each (Total Credits Hrs. $6*3=18$) 6 Papers of 2 Credits Hrs. each (Total Credits Hrs. $6*2=12$) 12 Papers of 1 Credits Hrs. each (Total Credits Hrs. $12*1=12$)	42
4	Discipline Specific Compulsory Course (DSC)	4 Papers of 2 Credits Hrs. each (Total Credits Hrs. $4*2=8$)	08
5	Discipline Specific Elective (DSE)	8 Papers of 3 Credits Hrs. each (Total Credits Hrs. $8*3=24$) 4 Papers of 2 Credits Hrs. each (Total Credits Hrs. $4*2=8$) 12 Papers of 1 Credits Hrs. each (Total Credits Hrs. $12*1=12$)	44
	Total Credits Hrs		120

Bachelor of Science in Computer Science (B. Sc. CS) Programme
Under Choice Based Credit, Grading and Semester System
Curriculum Framework
(Implemented during Academic Year 2022-23)

FIRST YEAR
Semester I & II (2022-2023)

Sr No	Semester I	Subject code	Total Marks	Credits	Sr No	Semester II	Subject code	Marks	Credits
	Ability Enhancement Compulsory Course (AECC)					Ability Enhancement Practical Compulsory Course (AECC)			
1	AECC-1 Communication Skills	2111UIS CS	60-40 100	3	1	AECC-2 IT platforms, Tools and Practices	2121UISTP	60-40 100	3
	Skill Enhancement Compulsory Course (SEC)					Skill Enhancement Compulsory Course (SEC)			
2	SEC-1 Web Programming-I	2112UIS WP	60-40 100	2	2	SEC-2 Web Programming – II	2122UISWP	60-40 100	2
	Web Programming-I Practical	2112UIS WPP	50	2		Web Programming – II Practical	2122UISWP P	50	2
	Core Course (CC)					Core Course (CC)			
3	CC-1 Introduction to Programming	2113UIS IP	60-40 100	2	3	CC-4 Programming and Application Development in Python	2123UISDP	60-40 100	2

	Introduction to Programming Practical	2113UIS IPP	50	2		Programming and Application Development in Python Practical	2123UISDP P	50	2
4	CC-2 Fundamentals of Computers and Electronics	2114UIS CE	60-40 100	2	4	CC-5 Object Oriented Programming	2124UISOP	60-40 100	2
	Fundamentals of Computers and Electronics Practical	2114UIS CEP	50	2		Object Oriented Programming Practical	2124UISOP P	50	2
5	CC-3 Operating Systems	2115UIS OS	60-40 100	2	5	CC-6 Database Management Systems - I	2125UISDB	60-40 100	2
	Operating Systems Practical	2115UIS OSP	50	2		Database Management Systems – I Practical	2125UISDB P	50	2
	<i>Discipline Specific Compulsory Course (DSC)</i>					<i>Discipline Specific Compulsory Course (DSC)</i>			
6	DSC-1 Discrete Mathematics-I	2116UIS DM	60-40 100	3	6	DSC-2 Discrete Mathematics - II	2126UISDM	60-40 100	3
	Total			22		Total			22

SECOND YEAR-Computer Science
(Implemented during Academic year 2022-23)

Sr. No.	Semester III	Subject code	Total Marks	Credits	Sr. No.	Semester IV	Subject code	Total marks	Credits
	Discipline Specific Compulsory Course (DSC)					Discipline Specific Compulsory Course (DSC)			
1	DSC-3 Descriptive Statistics	2231UISST	60-40 100	3	1	DSC-4 Probability Theory	2241UISPT	60-40 100	3
	Skill Enhancement Compulsory Course (SEC)					Skill Enhancement Compulsory Course (SEC)			
2	SEC-3 Core Java	2232UISCJ	60-40 100	2	2	SEC-4 Design Patterns	2242UISDP	60-40 100	2
	Core Java Practical	2232UISCJP	50	2		Design Patterns Practical	2242UISDPP	50	2
	Core Course (CC)					Core Course (CC)			
3	CC-7 Data Structures	2233UISDS	60-40 100	2	3	CC-10 Design and Analysis of Algorithms	2243UISDA	60-40 100	2
	Data Structures Practical	2233UISDSP	50	2		Design and Analysis of Algorithms Practical	2243UISDAP	50	2
4	CC-8 Computer Networks	2234UISCN	60-40 100	2	4	CC-11 System Programming	2244UISSP	60-40 100	2
	Computer Networks Practical	2234UISCNP	50	2		System Programming Practical	2244UISSPP	50	2
5	CC-9 Database Management Systems- II	2235UISDB	60-40 100	2	5	CC-12 Software Engineering	2245UISSE	60-40 100	2
	Database Management Systems- II Practical	2235UISDBP	50	2		Software Engineering Practical	2245UISSEP	50	2
	Discipline Specific Elective (DSE) CHOOSE ANY ONE					Discipline Specific Elective (DSE) CHOOSE ANY ONE			
6	Advanced Web Programming – I	2236UISAW	60-40 100	2	6	Advanced Web Programming - II	2246UISAW	60-40 100	2

Advanced Web Programming – I Practical	2236UISAWP	50	2		Advanced Web Programming – II Practical	2246UISAWP	50	2
Hybrid Mobile Application Development – I	2236UISMD	60-40 100	2		Hybrid Mobile Application Development - II	2246UISMD	60-40 100	2
Hybrid Mobile Application Development – I Practical	2236UISMDP	50	2		Hybrid Mobile Application Development – II Practical	2246UISMDP	50	2
Computer Graphics and Animation	2236UISCG	60-40 100	2		Advanced Java	2246UISAJ	60-40 100	2
Computer Graphics and Animation Practical	2236UISCGP	50	2		Advanced Java Practical	2246UISAJP	50	2
Embedded Systems	2236UISES	60-40 100	2		Multimedia Systems	2246UISMS	60-40 100	2
Embedded Systems Practical	2236UISESP	50	2		Multimedia Systems Practical	2246UISMSP	50	2
Theory of Computation - I	2236UISTC	60-40 100	2		Internet of Things	2246UISIT	60-40 100	2
Theory of Computation - I Practical	2236UISTCP	50	2		Internet of Things Practical	2246UISITP	50	2
					Theory of Computation - II	2246UISTC	60-40 100	2
					Theory of Computation - II Practical	2246UISTCP	50	2
	TOTAL		23		TOTAL			23

THIRD YEAR-Computer Science
(Implemented during Academic Year 2022-2023)

Sr No	Semester V	Subject code	Total Marks	Credits	Sr. No.	Semester VI	Subject code	Total Marks	Credits
	Core Course (CC)					Core Course (CC)			
1	CC 13: Data and Network Security	2251UCSDN	60-40 100	3	1	CC 16: Building blocks of Artificial Intelligence	2261UCSBA	60-40 100	3
	Data and Network Security Practical	2251UCSPR	50	1		CC 16: Building blocks of Artificial Intelligence Practical	2261UCSPR	50	1
2	CC 14: Optimization Techniques	2252UCSOT	60-40 100	3	2	CC 17: Data Science	2262UCSDS	60-40 100	3
	CC 14: Optimization Techniques Practical	2252UCSPR	50	1		CC 17: Data Science Practical	2262UCSPR	50	1
	Discipline Specific Elective (DSE)					Discipline Specific Elective (DSE)			
	Discipline Specific Elective (DSE-5) (Any ONE)					Discipline Specific Elective (DSE-7) (Any ONE)			
3	Web Services	2253UCSWS	60-40 100	3	3	Cloud Computing	2263UCSCC	60-40 100	3
	Web Services Practical	2253UCWSPR	50	1		Cloud Computing Practical	2263UCSCCPR	50	1
4	Ethical Hacking	2253UCSEH	60-40 100	3	4	Cyber Forensics	2263UCSCF	60-40 100	3

	Ethical Hacking Practical	2253UCSEHPR	50	1		Cyber Forensics Practical	2263UCSCF PR	50	1
5	Introduction to Big Data Analytics	2253UCSBD	60-40 100	3					
	Introduction to Big Data Analytics Practical	2253UCSBDPR	50	1					
	Discipline Specific Elective (DSE-6) (Any ONE)					Discipline Specific Elective (DSE-8) (Any ONE)			
6	Wireless Sensor Networks	2254UCSWN	60-40 100	3	5	Mobile and Pervasive Computing	2264UCSMP	60-40 100	3
	Wireless Sensor Networks Practical	2254UCSWNPR	50	1	11	Mobile and Pervasive Computing Practical	2264UCSMPPR	50	1
7	Game Programming	2254UCSGP	60-40 100	3	6	Digital Image Processing	2264UCSDI	60-40 100	3
	Game Programming Practical	2254UCSGPPR	50	1	12	Digital Image Processing Practical	2264UCSDIPR	50	1
8	Web Technologies	2254UCSWT	60-40 100	3					
	Web Technologies Practical	2254UCSWTPR	50	1					
17	Core 15: Project Implementation	2255UCSPI	100	4	13	Core 18: Project Implementation	2265UCSPI	100	4
	TOTAL			20		TOTAL			20

Nagindas Khandwala College (Autonomous)

**Syllabus and Question Paper Pattern
of Courses of**

Bachelor of Science Computer Science Programme

First Year

Semester I and II

Under Choice Based Credit, Grading and Semester System

(Implemented during Academic Year 2022-2023)

Bachelor of Science in Computer Science (B.Sc. CS) Programme

Under Choice Based Credit, Grading and Semester System

Curriculum Framework

(Implemented during Academic year 2022-2023)

FIRST YEAR

Sr No	Semester I	Subject code	Total Marks	Credits	Sr No	Semester II	Subject code	Marks	Credits
	Ability Enhancement Compulsory Course (AECC)					Ability Enhancement Practical Compulsory Course (AECC)			
1	AECC-1 Communication Skills	2111UISCS	60-40 100	3	1	AECC-2 IT platforms, Tools and Practices	2121UISTP	60-40 100	3
	Skill Enhancement Compulsory Course (SEC)					Skill Enhancement Compulsory Course (SEC)			
2	SEC-1 Web Programming-I	2112UISWP	60-40 100	2	2	SEC-2 Web Programming –II	2122UISWP	60-40 100	2
	Web Programming-I Practical	2112UISWP P	50	2		Web Programming – II Practical	2122UISWP P	50	2
	Core Course (CC)					Core Course (CC)			
3	CC-1 Introduction to Programming	2113UISIP	60-40 100	2	3	CC-4 Programming and Application Development in Python	2123UISDP	60-40 100	2
	Introduction to Programming Practical	2113UISIPP	50	2		Programming and Application Development in Python Practical	2123UISDP P	50	2

4	CC-2 Fundamentals of Computers and Electronics	2114UISCE	60-40 100	2	4	CC-5 Object Oriented Programming	2124UISOP	60-40 100	2
	Fundamentals of Computers and Electronics Practical	2114UISCE P	50	2		Object Oriented Programming Practical	2124UISOP P	50	2
5	CC-3 Operating Systems	2115UISOS	60-40 100	2	5	CC-6 Database Management Systems - I	2125UISDB	60-40 100	2
	Operating Systems Practical	2115UISOS P	50	2		Database Management Systems – I Practical	2125UISDB P	50	2
	<i>Discipline Specific Compulsory Course (DSC)</i>					<i>Discipline Specific Compulsory Course (DSC)</i>			
6	DSC-2 Discrete Mathematics-I	2116UISDM	60-40 100	3	6	DSC-2 Discrete Mathematics - II	2126UISDM	60-40 100	3
	Total			22		Total			22

***Syllabus of Courses of
Bachelor of Science Computer Science Programme
at Semester I
(Implemented during Academic Year 2022-2023)***

1. COMMUNICATION SKILLS

(wef 2020-21)

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Seven Cs of Effective Communication and Understanding Communication	9
2	Emotional Intelligence, Etiquette and Mannerism, Writing Business Messages and Documents	9
3	Academic Skills	9
4	Leadership and Team Building, Decision Making and Negotiation	9
5	Business ethics, Stress and Time Management	9
	Total	45

Objectives:

By the end of the course, learners will be able to:

1. Develop effective listening skills in learner so as to enable them to comprehend instructions and become a critical listener
2. Develop effective oral skills so as to enable learner to speak confidently interpersonally as well as in large groups
3. Develop effective writing skills so as to enable learner to write in clear, concise, persuasive and audience centred manner
4. Demonstrate effective use of communication technology

Course Outcome:

After completing this course, learners will be able to:

CO1: Understand the concept, channels, objectives, methods and modes of communication.
(Understand)

CO2: Differentiate obstacles to communication in the business world. (Evaluate)

CO3: Sharpen the business correspondence, language and writing skills of the learner. (Remember)

CO4: Effectively use communication technology. (Apply)

CO5: Demonstrate effective presentation, visual communication and impress stage. (Analyse)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	<p>The Seven Cs of Effective Communication: Completeness, Conciseness, Consideration, Concreteness, Clarity, Courtesy, Correctness Understanding Communication: Nature and Scope of Communication, Methods of communication, Cross-cultural communication, Technology-enabled Business Communication</p>	9
2	<p>Emotional Intelligence: Meaning and definition, Need for emotional intelligence, Intelligence Quotient versus Emotional Intelligence Quotient, Components of Emotional Intelligence, Competencies of Emotional Intelligence, and Skills to develop Emotional Intelligence. Etiquette and Mannerism: Introduction, Professional Etiquette, Technology Etiquette. Writing Business Messages and Documents: Business Correspondence: Letter of inquiry, letter of order, letter of complaints, sales letter, business reports, resume writing</p>	9
3	<p>Resume writing: Introduction, Resume, Curriculum Vitae, Job Application or Cover Letter Professional Presentation: Planning a Presentation, Preparing the Presentation, Delivering the Presentation Job Interviews: Types of job Interviews, Preparatory Steps for Job Interviews, Interview Skill Tips, FAQ During Interviews Group Discussion: Difference between Group Discussion, Panel Discussion and Debate, Importance of Group Discussions, Traits, Types of Group Discussions, Individual Traits</p>	9
4	<p>Leadership and Team Building: Leader and Leadership, Leadership Traits, Culture and Leadership, Leadership Styles and Trends, Team Building, Types of Teams, Decision Making and Negotiation: Introduction to Decision Making, Steps for Decision Making, Decision Making Techniques, Negotiation Fundamentals, Negotiation Styles, Major Negotiation Concepts</p>	9
5	<p>Business ethics: Importance of business ethics, personal integrity at work place, computer ethics, corporate social responsibility Stress and Time Management: Stress, Sources of Stress, Ways to Cope with Stress, time management, prioritizing and procrastination</p>	9

Reference Books

1. Meenakshi Raman and Prakash Singh, Business Communication, Oxford University Press, 2nd Ed.
2. Aruna Koneru, Professional Communication, Tata McGraw Hill
3. M. S. Rao, Strategies for Improving Your Business Communication, Shroff Publishers and Distributors
4. Soft Skills: an Integrated Approach to Maximise Personality, Gajendra S. Chauhan, Sangeeta Sharma, Wiley India

2. DISCRETE MATHEMATICS I

*(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No	Modules	No. of Lectures
1	Set Theory and Logic of Compound Statements	9
2	Relations and Functions	9
3	Sequences and Mathematical Induction	9
4	Recurrence Relations	9
5	Elementary Number Theory	9
	Total	45

Course Objectives:

By the end of the course learner will be able to:

1. Construct truth table and test the validity of statements.
2. Determine the domain and range of a discrete or non-discrete function, graph functions, identify one-to-one functions, perform the composition of functions, find and/or graph the inverse of a function, and apply the properties of functions to application problems.
3. List the terms in a sequence, write a sequence in closed form, compute the sum of a finite sequence, compute the product of a finite sequence, and express sequences in terms of recursive or non-recursive forms.
4. Use elementary number theory including the divisibility properties of numbers to determine prime numbers and composites, the greatest common divisor, and the least common multiple; perform modulo arithmetic and computer arithmetic.
5. Recall sequences and mathematical induction.

Course Outcome:

After completing this course learner will be able to:

CO1: Remember theory of discrete objects, starting with relations and partially ordered sets. (Remember)

CO2: Understand recurrence relations, generating function and their applications. (Understand)

CO3: Apply well-ordering principle for integers and check correctness of algorithm. (Apply)

CO4: Recall types of functions and find their applications. (Analyze)

CO5: Locate maximum and minimum elements for Lattices. (Evaluate)

Detailed Syllabus:

Module	Topics	No of Lectures
1	Set Theory: Definitions and the Element Method of Proof, Properties of Sets, Disproof's, Algebraic Proofs, Boolean Algebras, Russell's Paradox and the Halting Problem. Logic of Compound Statements: Logical Form and Logical Equivalence, Conditional Statements, Valid and Invalid Arguments, Conditional Statements, CONVERSE, CONTRAPOSITIVE, AND INVERSE, Truth Tables of Compound Propositions.	9
2	Relations: Relations on Sets, Reflexivity, Symmetry, and Transitivity, Equivalence Relations, Partial Order Relations, Linear Ordering Hasse Diagrams, Maximum and Minimum elements, Lattices Functions: Definition of function. Domain, co domain and the range of a function. Direct and inverse images. Injective, surjective and bijective functions. Composite and inverse functions.	9
3	Sequences and Mathematical Induction: Sequences, Mathematical Induction, Strong Mathematical Induction and the Well-Ordering Principle for the Integers, Correctness of algorithms, defining sequences recursively	9
4	Recurrence Relations: Definition of recurrence relations, Formulating recurrence relations, solving recurrence relations- Back tracking method, Solving linear homogeneous recurrence relations with constant coefficients of degree two when characteristic equation has distinct roots and only one root, Applications- Formulate and solve recurrence relation for Fibonacci numbers, Tower of Hanoi	9
5	Elementary Number Theory: Introduction to Direct proofs, Rational Numbers, Divisibility, Quotient – Remainder theorem, Indirect Argument, Contradiction and Contraposition, Two Classical Theorems.	9

Reference Books

1. Discrete Mathematics with Applications, Sussana S. Epp, Cengage Learning, 4th Edition, 2010.
2. Discrete Mathematics, Schaum's Outline Series, Seymour Lipschutz, Tata McGraw Hill, 2007.
3. Discrete Mathematics and its Applications, Kenneth H. Rosen, Tata McGraw Hill, 2015.
4. Discrete Mathematical Structures, 6th Ed., Kolmann R. C. Busby, S. Ross, PHI, 2009.
5. Elements of Discrete Mathematics, 4th Ed., C. L. Liu, D. P. Mohapatra, Tata McGraw Hill, 2012.

3. INTRODUCTION TO PROGRAMMING

*(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No	Modules	No. of Lectures
1	Strategies for problem-solving	6
2	General problem-solving techniques and Algorithm complexity analysis	6
3	Pseudo Codes and Python	6
4	Pseudo Codes and Python (Conditional statements and Loops)	6
5	Searching and Sorting, Python, Project	6
	TOTAL	30

Course Objectives:

On completing this course learners will be able to:

1. To enhance the logical thinking
2. To develop problem solving skills
3. To introduce the basic programming concepts
4. write programs in Python and Scratch

Course Outcome:

After successful completion of this course, learners will be able to:

CO1: Demonstrate the need of problem solving skills and demonstrate the solution to the standard problems (understand)

CO2: Analyse whether the given strategy is suitable for a given problem (evaluate)

CO3: Explain the basic programming structures in Python (understand)

CO4: Compare and suggest the suitable statements in Python for a given problem (analyze and apply)

CO5: Create animations in Scratch for a selected problem. (Create)

CO6: Write and demonstrate the working of basic programmes in Python (Apply)

Detailed Syllabus:

Module	Topics	No of Lectures
1	<p>Strategies for problem-solving: Classic Puzzles: How to cross the river with fox, goose and corn, The sliding eight, sliding five, completing a Sudoku square, Quarrasi lock.</p> <p>Pseudo code, Algorithm and flow-chart creation for basic problems.</p>	6
2	<p>General problem solving techniques: Always have a plan, restate the problem, divide the problem, Start with What You Know, Reduce the problem, Look for analogies, experiment, Don't get frustrated Output method, Tracking down logical errors, Output patterns: Half of a square, count down by counting up, sideways triangle.</p> <p>Algorithm complexity analysis: Time and space complexity, asymptotic notations.</p>	6
3	<p>Pseudo Codes: convert character digits to integers, checksum validation, identification of positive or negative numbers, encode/decode a message, find mean/median/mode, tower of Hanoi.</p> <p>Python: Variables, Strings, Numeric Types, Typecasting, Python operators, Function: introduction, function definition, parameters, function call, recursions.</p>	6
4	<p>Pseudo Codes: finding the highest revenue customer, the iterative sum of list elements, Translator program, Hangman game, scientific calculator.</p> <p>Python: import statement, Branching (if, if-else, if-elif-else, switch-case structure), Looping (while, for, break and continue statement)</p>	6
5	<p>Searching and Sorting: Searching-Linear Search, Binary Search, Sorting-Bubble, Selection and Insertion Sort, Working with Sorted Lists-Maintaining Sorted Lists Exception handling: What is an exception, various keywords to handle exceptions such as try, catch, except, else, finally, raise.</p> <p>Python: List, slicing, Tuples, Dictionaries, Nested Elements, store, copy, search, append, sort operations.</p> <p>Project: Semester end project.</p>	6

Reference Books

1. Paul Gries, Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014
2. Michael Dawson, Python Programming for the Absolute Beginner, Paperback, Second Edition, Published November 8th, 2005 by Course Technology PTR
3. James Payne, Beginning Python: Using Python 2.6 and Python 3, Wiley India, 2010

Practical:

1. Write a Scratch program for (any one):
 - a. How to cross the river with fox, goose, and corn
 - b. The sliding eight
 - c. The sliding five
2. Perform following
 - a. Python and Jupyter lab setup.
 - b. Write a Python program for printing the patterns.
3. Write a Python program to check:
 - a. Whether the given number is odd or even
 - b. Whether the given year is a leap year
4. Write a Python program to convert:
 - a. Character digit to an integer.
 - b. A decimal number to binary or hexadecimal.
5. Write a Python program to check:
 - a. Whether the given string is a palindrome or not.
 - b. Whether the given number is a prime number or not.
6. Write a Python program to find:
 - a. The nth Fibonacci number.
 - b. Factorial of a given number.
7. Write a Python program to convert read a line of text and count
 - a. The number of letters
 - b. Number of words
8. Write a Python program to read a list of values and determines the quartiles.
9. Write a Python function to calculate space and time complexity.
10. Write Python functions to encode and decode a message.
11. Write a python function to find the sum of positive integers in a list
 - a. Iteratively
 - b. recursively
12. Write a function to implement
 - a. Sorting with Exception handling.
 - b. Searching with Exception handling.

4. FUNDAMENTALS OF COMPUTERS AND ELECTRONICS

(Implemented during Academic Year 2022-23)
(wef 2020-21)

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Computer Abstraction and Technology	6
2	Alphanumeric Codes and Binary Arithmetic	6
3	Boolean Algebra and Minimization	6
4	Combinational and Sequential Circuits	6
5	8085 Instruction Set Architecture	6
Total		30

Course objectives –

By the end of the course, learners will be able to:

- Explain the concept of computer systems and get acquainted with number systems.
- Understand Boolean Algebra and Minimization concepts.
- Understand different types of logic gates.
- Explain different types of digital circuits.
- Code in 8085 Assembly Language.

Course Outcome –

After completing this course learner will be able to:

CO1: Understand and perform conversion between different number systems. (Apply)

CO2: Understand and analyse different Boolean theorems and laws and implement it in reduction of logic expression. (Apply)

CO3: Analyse different logic gates, minimize the given logical expression and create circuits from it. (Analyse)

CO4: Understand the Combinational and Sequential circuits and its application in designing circuits. (Evaluate)

CO5: Learners will also be able to create code using 8085 Assembly language. (Create)

Detailed Syllabus:

Module	Topics	No of Lectures
1	<p>Computer Abstraction and Technology:</p> <p>Basic Structure of a Computer: Computer Types, Functional Units, Basic Operational Concepts, Performance, And Historical Perspective.</p> <p>Number System: Bits, bytes, analog system, digital system, binary number system, octal number system, hexadecimal number system. Number System Conversion: Conversion from one number system to another, floating point numbers, weighted codes, binary coded decimal, non-weighted codes, excess – 3 code, gray code</p>	6
2	<p>Alphanumeric Codes and Binary Arithmetic:</p> <p>Alphanumeric codes: ASCII Code, EBCDIC, ISCII Code, Hollerith Code, Morse Code, Teletypewriter (TTY), Error detection, and correction, Universal Product Code.</p> <p>Binary Arithmetic: Binary addition, Binary subtraction, Negative number representation, Subtraction using 1's complement and 2's complement, Binary multiplication and division, Arithmetic in octal number system, Arithmetic in hexadecimal number system, BCD and Excess – 3 arithmetic.</p>	6
3	<p>Boolean Algebra and Minimization:</p> <p>Boolean Algebra and Logic Gates: Introduction, Logic (AND OR NOT), Boolean theorems, Boolean Laws, De Morgan's Theorem, Perfect Induction, Reduction of Logic expression using Boolean Algebra, Deriving Boolean expression from given circuit, exclusive OR and Exclusive NOR gates, Universal Logic gates, Implementation of other gates using universal gates, Input bubbled logic, Assertion level. Min-term, Max-term and K-Maps: Introduction, min-terms and sum of min-term form, max-term and Product of max-term form, Reduction technique using k-maps – 2/3/4/5/6 variable K-maps, Grouping of variables in K-maps, K-maps for product of sum form, minimize Boolean expression using K-map and obtain K-map from Boolean expression.</p>	6
4	<p>Combinational and Sequential Circuits:</p> <p>Combinational circuit: Introduction, Multi-input and multi-output Combinational circuits, Introduction, Adder, BCD Adder, Excess – 3 Adder, Binary subtractor, BCD Subtractor, Multiplier, Comparator, Multiplexer, Demultiplexer, Decoder, Encoders. Sequential circuits: Flip- Flop(SR, JK), Shift Registers(serial– in serial–out, serial–in parallel–out , parallel–in parallel–out).</p> <p>Counters: Synchronous and Asynchronous</p>	6

5	<p>Instruction Set Architecture: Introduction: Microprocessor, Microprocessor Instruction Set and Computer Languages, From Large Computers to Single-Chip Microcontrollers, Applications</p> <p>Introduction to 8085 Assembly Language Programming: 8085 Programming Model, Instruction Classification, Data and Storage, Writing, assembling and Execution of a simple program, Writing and Assembling Program.</p>	6
---	--	---

Reference Books

1. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012.
2. Patterson and Hennessy, Computer Organization and Design, Morgan Kaufmann, ARM Edition, 2011.
3. R P Jain, Modern Digital Electronics, Tata McGraw Hill Education Pvt. Ltd. , 4th Edition, 2010.
4. Ramesh Gaonkar, Microprocessor Architecture, programming and Applications with the 8085, PENRAM, 5th Ed., 2012.

Practical:

1. Study and verify the truth table of various logic gates.
2. Verify De-Morgan's Law
3. Simplify given Boolean expression and realize it.
4. Design and verify a half/full adder
5. Design and verify half/full subtractor
6. Design a 2 bit comparator using combinational circuits.
7. Perform the following Operations related to memory locations:
 - a. Store the data byte 32H into memory location 4000H.
 - b. Exchange the contents of memory locations 2000H and 4000H
8. Arithmetic operations:
 - a. Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.
 - b. Add the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.
 - c. Multiply the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.
9. Complement operations:
 - a. Find the 1's complement of the number stored at memory location 4400H and store the complemented number at memory location 4300H.

- b. Find the 2's complement of the number stored at memory location 4200H and store the complemented number at memory location 4300H.

10. Logical operations:

- a. AND the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.
- b. OR the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H
- c. X-OR the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.

5. OPERATING SYSTEMS

*(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1	Introduction and Overview of Operating System	6
2	Process Management-I	6
3	Process Management-II	6
4	Memory Management	6
5	Storage Management	6
	Total	30

Course Objective:

By the end of the course learner will be able to:

1. Understand the services provided by and the design of an operating system.
2. Recognise what a process is and how processes are synchronized and scheduled.
3. Differentiate between different approaches to memory management.
4. Learn virtual memory and secondary memory management.
5. Learn different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system

Course Outcome:

After completing this course learner will be able to:

- CO1: Understand the structure of OS and basic architectural components involved in OS design.
(Understand)
- CO2: Analyze and design the applications to run in parallel either using process or thread models of different OS (Apply)

CO3: Differentiate various device and resource management techniques for time sharing and distributed systems. (Evaluate)

CO4: Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system (Understand)

CO5: Conceptualize the components involved in designing a contemporary OS (Create)

Detailed Syllabus:

Module	Topics	No of Lectures
1	<p><u>OverView</u> Introduction: Definition of Operating system, Operating System's role, Operating-System Operations, Functions of Operating System, Computing Environments</p> <p>Operating-System Structures: Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, Operating-System Structure</p>	6
2	<p><u>Process Management-I</u> Processes: Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication Threads: Overview, Multicore Programming, Multithreading Models</p> <p>Process Synchronization: General structure of a typical process, race condition, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors</p>	6
3	<p><u>Process Management-II</u> CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, SRTF, Priority, RR, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling), Thread Scheduling</p> <p>Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock</p>	6
4	<p><u>Memory Management</u> Main Memory: Background, Logical address space, Physical address space, MMU, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table</p> <p>Virtual Memory: Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing</p>	6

5	<p><u>Storage Management</u></p> <p>Mass-Storage Structure: Overview, Disk Structure, Disk Scheduling, Disk Management</p> <p>File-System Interface: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing</p>	6
---	--	---

Reference Books

1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley, 9th Edition, 2012
2. Achyut S. Godbole, Atul Kahate, Operating Systems, 2nd Ed., Tata McGraw Hill, 2009
3. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 4e Fourth Edition, Pearson Education, 2016
4. Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, Tata McGraw Hill Education Private Limited, 2013

Practical:

1. Installation of **Linux operating system (Red-Hat / Ubuntu) on virtual machine:** Configuring Linux system – creating user accounts – configuring networking.
2. Installation of **Windows operating system on virtual machine:** Configuring Windows - Service pack installation - OS updation - software installation – network configuration.
3. Linux commands: Linux commands: Working with Directories:
 - a) pwd, cd, absolute and relative paths, ls, mkdir, rmdir
 - b) file, touch, rm, cp, mv, rename, head, tail, cat, tac, more, less, strings, chmod Linux commands: Working with files:
 - a) ps, top, kill, pkill, bg, fg
 - b) grep, locate, find, locate
 - c) date, cal, uptime, w, whoami, finger, uname, man, df, du, free, whereis, which
 - d) Compression: tar, gzip
4. Working with Linux Desktop and utilities
 - a) The vi editor.
 - b) Graphics
 - c) Terminal
 - d) adjusting display resolution
 - e) Using the browsers
 - f) Configuring simple networking
 - g) Creating users and shares
5. Windows (DOS) Command-s
 - a) Date, time, prompt, md, cd, rd, path, color
 - b) Chkdsk, copy, xcopy, format, cls, defrag, del, move
 - c) Diskcomp, diskcopy, diskpart, doskey, echo
 - d) Edit, fc, find, rename, set, type, ver, attrib, deltree, tree
 - e) ipconfig, ping

6. Working with Windows Desktop and utilities a) Notepad b) Wordpad c) Paint d) Taskbar e) adjusting display resolution f) Using the browsers g) Configuring simple networking h) Creating users and shares
7. **Identification of Hardware components** - motherboard, processor, memory (DDR3,DDR4), HDD, DVD writer, SMPS - NIC– UPS – online & offline.
8. **Identification of connectors and slots in motherboard.** Ports - serial, parallel, USB, PS2, audio ports.
9. **Installing and connecting various Devices** - Printers, scanners, Ethernet, Blue tooth, wireless, mobile, modem, projector
10. Installation of ownCloud client and connecting with ownCloud server using demo.owncloud.org and Study of Google Drive

Self-Assessment Exercise

Exercise1

1. Use MS-DOS to create the above directory structure.
2. Create 3 files containing a list of songs for three of the artists. Save the files using no more than 8 letters for the filename.
3. Copy the files into the appropriate directory.
4. Ensure that you can navigate from the root to a given sub-directory and that you can copy a file to a given directory.
5. Remove the directory named Country.

Exercise2

1. Copy FILE1.TXT and make a file2.txt, file3.dat, file4.xyz, file5.txt. Pay attention to the names and extensions of the files.
2. List out the directory and see ALL the newly created files.
3. Now you just want to see a list of the files that have TXT as their extension.
4. Make a copy of file4.xyz and call it first.dat
5. Create a subdirectory under the directory you're currently in. Call this new directory TESTDIR
6. Copy all the files with an extension of DAT to the new subdirectory.
7. Change directories to the newly created directory and list all the files in there.
8. Delete file3.dat.
9. Go back up to the parent directory of the current directory you are in.
10. List all the file with an extension of DAT in the current directory and the one you just created. Use only one command to do so.
11. Change the name of file1.txt to file6.txt.
12. Delete the directory you created (TESTDIR). Are you able to do this ? Why or Why not ?
13. Change directories to the TESTDIR directory. Delete the remaining file. Change directories up to the parent directory (up one level) and now delete the directory.
14. List all the files that start with FI and have an extension of TXT.
15. List all the files that start with the word FILE, then a single digit an extension of TXT.

Exercise3

1. Create directory with your name.
2. Create two new subdirectories labelled: DIR1 and DIR2 in the YOUR directory. Create two subdirectories labelled dir11 and dir12 in the DIR1 directory. Create a subdirectory labeled dir21 in the DIR2 directory.
In the DIR1 directory, create a text file named myfile.txt, and write your firstname, lastname in this file.
3. Copy the file myfile.txt to the DIR2 directory.
4. Create a file myfile2.txt using notepad from MSDOS and mention today's date and time in the file.
5. Display the content of the myfile2.txt.
6. Copy the content of myfile2.txt into FILE.txt.
7. Move the file FILE.txt to dir21.
8. Change the prompt with \$.
9. Display the current path of the directory.
10. Display all the directories with their files.

Exercise 4

1. Create a directory with your name and rollno.
2. Change the current path to the created directory.
3. Create a file with 10 names in the directory.
4. Display the content of the file.
5. Display the content of the file in sorted order.
6. Open the file using vi editor and add more 5 names in the file.
7. Create 1 more directory and copy the above created file in the created directory.
8. Print the content of the file in reverse order.
9. Print the total words from a file.
10. Search the content with 'a' in file

6. WEB PROGRAMMING I

(Implemented during Academic Year 2022-23)
(wef 2020-21)

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1	Internet and world wide web, HTML5	6
2	HTML5 Components, Style sheets	6
3	JavaScript	6
4	Document and its associated objects	6
5	JQuery and Bootstrap	6
	Total	30

Course Objectives:

By the end of the course, learners will be able to:

1. Identify and learn the Internet World with working of a website using HTML.
2. Identify the creation of dynamic websites using different components of HTML.
3. Define and describe the javascript usage.
4. State and Explain the different document object models.
5. Explain jQuery and bootstrap components.

Course Outcome:

After completing this course learners will be able to:

- CO1: Apply a structured approach to identifying needs, interests, and functionality of a website.
(Apply)
- CO2: Design dynamic websites that meet specified needs and interests.(Create)
- CO3: Design appropriate HTML, CSS, and JavaScript code from public repositories of open-source and free scripts that enhances the experience of site visitors.(Create)
- CO4: Analyze the existing HTML, CSS, and JavaScript code to extend and alter its functionality, and to correct errors and cases of poor practice. (Analyse)
- CO5: Create a website which is functional with all the basics and advanced HTML,CSS, Javascript alongwith jQuery and Bootstrap.(Apply)

Detailed Syllabus:

Module	Topics	No of Lectures
1	<p>Internet and world wide web: What is Internet? Introduction to internet and its applications, E-mail, telnet, FTP, e-commerce, video conferencing, e-business. Internet service providers, domain name server, internet address,</p> <p>World Wide Web (WWW): World Wide Web and its evolution, uniform resource locator (URL),</p> <p>Browsers – internet explorer, Netscape navigator, opera, Firefox, chrome, Mozilla. Search engine, HTTP protocol</p> <p>HTML5: Introduction, Why HTML5? Formatting text by using tags, using lists and backgrounds, Creating hyperlinks and anchors. Style sheets, CSS formatting text using style sheets, formatting paragraphs using style sheets</p>	6
2	<p>HTML5 Page layout and navigation: Creating navigational aids: planning site organization, creating text based navigation bar, creating graphics based navigation bar, creating graphical navigation bar, creating image map, redirecting to another URL, creating division based layouts: HTML5 semantic tags, creating divisions, creating HTML5 semantic layout, positioning and formatting divisions.</p> <p>HTML5 Tables, Forms and Media: Creating tables: creating simple table, specifying the size of the table, specifying the width of the column, merging table cells, using tables for page layout, formatting tables: applying table borders, applying background and foreground fills, changing cell padding, spacing and alignment, creating user forms: creating basic form, using check boxes and option buttons, creating lists, additional input types in HTML5, Incorporating sound and video: audio and video in HTML5, HTML multimedia basics, embedding video clips, incorporating audio on web page. HTML web storage.</p> <p>Style Sheets: What are style sheets?, Why are style sheets valuable?, Different approaches to style sheets, Using Multiple approaches, Linking to style information in separate file, ,Setting up style information, Using the tag, Embedded style information.</p>	6
3	<p>Java Script: Introduction, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Security, Operators: Assignment Operators, Comparison Operators, Arithmetic Operators, % (Modulus), ++(Increment), -- (Decrement), -(Unary Negation), Logical Operators, Short-Circuit Evaluation,</p>	6

	String Operators, Special Operators, ?: (Conditional operator), , (Comma operator), delete, new, this, void Statements: Break, comment, continue, delete, do...while, export, for, for...in, function, if...else, import, labelled, return, switch, var, while, with. Core JavaScript (Properties and Methods of Each) : Array, Boolean, Date, Function, Math, Number, Object, String, regExp	
4	Document and its associated objects: document, Link, Area, Anchor, Image, Applet, Layer Events and Event Handlers : General Information about Events, Defining Event Handlers, event, onAbort, onBlur, onChange, onClick, onDbClick, onDragDrop, onError, onFocus, onKeyDown, onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onMove, onReset, onResize, onSelect, onSubmit, onUnload	6
5	JQUERY: Introduction of jquery, syntax, selectors, events, effects, jquery html and css methods. Bootstrap: Introduction of basic bootstrap, installation, using bootstrap grid, using base css	6

Reference Books

1. HTML5 Black Book: Covers CSS3, JAVASCRIPT, XML,XHTML, AJAX, PHP and JQUERY DreamTech Press.
2. Learning bootstrap : Aravind Shenoy,Ulrich Sossou, Packt publishing
3. Learning Jquery: Jonathan Chaffer, Karl Swedberg, Packt publishing
4. JavaScript 2.0: The Complete Reference, Thomas Powell and Fritz Schneider, Tata Mcgraw Hill
5. HTML5 Step by Step, Faithe Wempen, Microsoft Press

Practical:

1. Use of Basic Tag
 - a. Design a webpage using different text formatting tags.
 - b. Design a webpage using hyperlinks to different pages and allow navigation between webpages (atleast 3 web pages)
 - c. Design a webpage demonstrating all Style sheet type
2. Imagemaps, Tables, Forms and Media
 - a. Design a webpage with Imagemaps.
 - b. Design a webpage demonstrating different semantics
 - c. Design a webpage with different tables. Design a webpages using table so that the content appears well placed.
 - d. Design a webpage with a Registration / Feedback form that uses all types of controls.
 - e. Design a webpage embedding with multimedia features.

3. JavaScript

- a. Using JavaScript design, a webpage that prints factorial / Fibonacci series / any given series.
 - b. Design a form and validate all the controls placed on the form using Java Script.
 - c. Write a JavaScript program to display all the prime numbers between 1 and 100.
 - d. Write a JavaScript program to accept a number from the user and display the sum of its digits.
 - e. Write a program in JavaScript to accept a sentence from the user and display the number of words in it. (Do not use split () function).
 - f. Write a java script program to design simple calculator.
 - g. Write a java script program to validate the form.
4. Control and looping statements and Java Script references
 - a. Design a web page demonstrating different conditional statements
 - b. Design a web page demonstrating different looping statements.
 5. Design a web page demonstrating different Core JavaScript references (Array, Boolean, Date, Function, Math, Number, Object, String, regExp).
 6. Design a web page demonstrating key Events and mouse events
 7. Design a web page demonstrating jquery events and effects.
 8. Design a web page demonstrating jquery html and css methods
 9. Design a web page demonstrating bootstrap
 10. Demonstrate program on HTML web storage.

Evaluation Scheme

Test– 20 Marks

It will be conducted either as a written test or using any open source learning management system such as Moodle (Modular object-oriented dynamic learning environment) or a test based on an equivalent online course on the contents of the concerned course (subject) offered by or build using MOOC (Massive Open Online Course) platform or a written test conducted in the class room.

Assignments/Presentation: 15 marks

5 Marks - Active participation in routine class instructional deliveries:

Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

Semester End Examination- 60 Marks

Duration - 2 Hours.

Theory question paper pattern:-

All questions are compulsory.		
Question	Based on	Marks
Q1	Unit 1, 2, 3, 4, 5	4 out of 5 questions (05 marks each)
Q2	Unit 1, 2, 3, 4, 5	2 out of 3 questions (07 marks each)
Q3	Unit 1, 2, 3, 4, 5	2 out of 3 questions (08 marks each)
Q4	Based on multiple Units	1 out of 2 questions (10 marks)

Practical Examination – 50 marks (Duration: 2 Hours)

- Each practical course carries 50 Marks : 40 marks + 05 marks (journal)+ 05 marks(viva)
- Minimum 75% practical from each core/allied course are required to be completed and written in the journal.

(Certified Journal is compulsory for appearing at the time of Practical Examination)

***Syllabus of Courses of
Bachelor of Science in Computer Science Programme
at Semester II
(Implemented during Academic Year 2022-23)***

**1. PROGRAMMING AND APPLICATION DEVELOPMENT IN
PYTHON
(wef 2020-21)**

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1.	GUI Programming in Python (using Tkinter/wxPython/Qt)	6
2.	Database connectivity in Python	6
3.	Python File Input-Output	6
4.	Regular Expressions, Python and the Web	6
5	Network connectivity, Project	6
Total		30

Course Objectives:

By the end of the course, learners will be able to:

1. Design UI and program python UI applications.
2. Connect database in a python program.
3. Read and write files and file operations.
4. Write a program on regular expressions.
5. Program web applications and implement web scraping in python .
6. Learn network connectivity.

Course Outcome :

After completing this course, learner will be able to:

CO1: Implement UI Applications using Python Tkinter.(Apply)

CO2: Apply the Knowledge of Database Connection in Python Application.(Apply)

CO3: Understand the working of different File Operations.(Understand)

CO4: Analyse the data obtained using Web Scraping and Develop Network Connectivity Applications.(Analyse)

CO5: Create a small application showing the implementation of topics learned.(Create)

Detailed Syllabus:

Module	Topics	No of Lectures
1	GUI Programming in Python (using Tkinter/wxPython/Qt): What is GUI, Advantages of GUI, Introduction to GUI library, Layout management, events, and bindings, fonts, colors, drawing on canvas (line, oval, rectangle, etc.). Widgets such as: frame, label, button, checkbutton, entry, listbox, message, radiobutton, text, spinbox etc.	6
2	Database connectivity in Python: Installing mysql connector, accessing connector module using connect, cursor, execute & close functions, reading single & multiple results of query execution, executing different types of statements, executing transactions, understanding exceptions in database connectivity.	6
3	Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their Problem solving applications. Python Modules and Packages, Lambda function.	6
4	Regular Expressions: Concept of regular expression, various types of regular expressions, using match function. Python and the Web: Screen scraping, Web Scraping (Using Scrapy/Selenium/BeautifulSoup etc.	6
5	Network connectivity: Socket module, creating server-client programs, sending Email, reading from URL Project: Semester end project.	6

Reference Books

1. Paul Gries, Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014
2. Python Programming for the Absolute Beginner by Michael Dawson Paperback, Second Edition, 472 pages Published November 8th, 2005 by Course Technology PTR
3. James Payne, Beginning Python: Using Python 2.6 and Python 3, Wiley India, 2010

Practical:

1. Programs to read and write files.
2. Programs with iterables and iterators.
3. Program to connect to a DB and execute various SQL queries.
4. Program to demonstrate the use of regular expressions.
5. Program to show draw shapes & GUI controls. (Mini project including database connectivity)
6. Program to create server-client and exchange basic information.
7. Use Scrapy/Selenium/BeautifulSoup for web mining.
8. Program to send email & read contents of URL.
9. Create a registration and login application using Python Tkinter. Registered data should be stored in the database and login data should be compared with the registered data to check if the user is registered or not.

2. OBJECT ORIENTED PROGRAMMING
at Semester II
(Implemented during Academic Year 2022-23)
(wef 2020-21)
Modules at a Glance

Sr. No	Modules	No. of Lectures
1	Object oriented paradigm	6
2	Creating class diagrams and Generate Blueprints using Classes	6
3	Encapsulation of Data	6
4	Inheritance and Specialization	6
5	Interfaces, Multiple Inheritance, and Composition, Visualization	6
	TOTAL	30

Course Objectives –

By the end of the course learner will be able to:

1. Study the principles of object-oriented paradigms.
2. Understand how real-world objects can become part of fundamental elements in the code.
3. Understand the difference between classes, prototypes, and instances.
4. Learn to organize data in the blueprints and create a hierarchy of blueprints that generate objects.
5. Develop basic object-oriented code using object-oriented languages.
6. Implement all concepts of OOP in Program development.

Course Outcome:

After completing this course learner will be able to:

CO1: Define various concepts of Object Oriented Programming. (Remember)

CO2: Identify real-world objects and design Class diagram to organize data.(Understand)

CO3: Generate blueprints to create objects. (Create)

CO4: Apply standards and principles to write truly readable code. (Apply)

CO5: Develop and test basic programs. (Understand)

CO6: Demonstrate the concepts of object-oriented design, polymorphism, information hiding, and inheritance. (Apply)

Detailed Syllabus:

Module	Topics	No. of Lectures
1.	<p>Introduction: A look at Procedure-Oriented Programming, OOP Paradigm, Basic concepts of OOP, Benefits of OOP, OOP Languages, Applications of OOP.</p> <p>Objects Everywhere: Recognizing objects from nouns, Generating blueprints for objects, Recognizing attributes/behaviour, Recognizing actions from verbs – methods, Organizing the blueprints – classes using UML diagrams.</p>	6
2.	<p>Classes and Objects: Understanding classes and objects, Understanding constructors and destructors, Creating classes, Customizing constructors, Customizing destructors, Creating objects of classes</p>	6
3.	<p>Encapsulation of Data: Understanding the different members of a class, Protecting and hiding data, Adding attributes to a class, Adding properties to a constructor function, Hiding data using prefixes, Using access modifiers, Using property getters and setters, Using methods to add behaviours to classes.</p>	6
4.	<p>Inheritance and Specialization: Using classes to abstract behaviour, Understanding inheritance, Polymorphism, Understanding method overloading and overriding, Understanding operator overloading, Taking advantage of polymorphism.</p>	6
5.	<p>Interfaces, Multiple Inheritance, and Composition: Understanding the requirement to work with multiple base classes, Working with multiple inheritance, Declaring base classes for multiple inheritance, Declaring classes that override methods, Declaring a class with multiple base classes, Working with objects of classes that use multiple inheritance, Working with abstract base classes.</p> <p>Data Visualization: Introduction, History, Importance, Benefits, Data visualization in today's world, How it is used, How it works</p>	6

Reference Books

1. Learning Object-Oriented Programming, Gaston C. Hillar, Packt, 1st Edition, 2015.
2. Python 3 Object-oriented Programming, Dusty Phillips, Packt, 2nd Edition, 2015.
3. The Object-Oriented Thought Process, Matt Weisfeld, 3rd Edition, 2009
4. https://www.sas.com/en_in/insights/big-data/data-visualization.html
5. <https://realpython.com/python3-object-oriented-programming/#how-to-define-a-class-in-python>

Practical:

1. Creating Class diagrams with the abstract, subclasses, their attributes and methods.
2. Defining and using a class.
3. Defining methods with and without attributes in a class.
4. Creating and using constructor and Destructor.
5. Using property getters and setters.
6. Implementing various forms of Inheritance.
7. Implementing Polymorphism by Overloading Overriding methods.
8. Implementing the concept of Operator Overloading.
9. Implementing abstract classes and interfaces.
10. Implementing the concept of Composition.

3. DATABASE MANAGEMENT SYSTEMS I
at Semester II
(Implemented during Academic Year 2022-23)
(wef 2020-21)

Modules at a Glance

Sr. No	Modules	No. of Lectures
1	Introduction and Data Models	6
2	ER Diagram, Relational Database Design, Relational Algebra	6
3	Joining Tables, Sub queries, Views	6
4	Transaction and Concurrency Control, Crash Recovery	6
5	Introduction to SQL, View, Functions	6
	TOTAL	30

Course Objective:

By the end of the course, learners will be able to:

- Understand the basic concepts and the applications of database systems.
- Master the basics of SQL and construct queries using SQL.
- Familiar with the basic issues of transaction processing and concurrency control.
- Understand the concept of Normalization, Transaction and Concurrency Control.
- Understand the concept of PL/SQL procedures and construct.

Course Outcome:

After completing this course, learners will be able to:

CO1: Explain the basic elements of a relational database management system(Understand)

CO2: Design entity relationship and convert entity relationship diagrams into RDBMS and formulate SQL queries on the respect data into RDBMS and formulate SQL queries on the data. (Analyze, Create)

CO3: Apply various Normalization techniques (Apply)

CO4: Perform PL/SQL programming using concept of Cursor Management, Error Handling, Package and Triggers (Create)

CO5: Understand the principles of Transaction Processing & Locking using the concept of

Concurrency control. (Understand)

Detailed Syllabus:

Module	Topics	No. of Lectures
1.	<p>Introduction to DBMS: Definition, Overview of DBMS, Advantages of DBMS, Purpose of Database Systems, View of Data, Database Languages, , Levels of abstraction, Data independence, DBMS Architecture, Limitation of DBMS, Introduction of NoSQL, Comparison between DBMS and RDBMS.</p> <p>Data Models: The importance of data models, Basic building blocks, Business rules, Client/Server Architecture, Object Based Logical Model, Record Based Logical Model (relational, hierarchical, network), Degrees of data abstraction.</p>	6
2.	<p>ER-Diagram : Database design and ER Model: Overview, ER-Model, Constraints, ER Diagrams, ERD Issues, weak entity sets, Codd’s rules, Relational Schemas</p> <p>Relational Database design: Features of good relational database design, atomic domain and Normalization (1NF, 2NF, 3NF, BCNF).</p> <p>Relational Algebra: Operations - selection, projection, Set operations- union, intersection, difference, cross product, Joins –conditional, equi join and natural joins, division operator</p>	6
3.	<p>Joining tables: Inner, Equi , Outer , Cross and Self join</p> <p>Sub queries: Sub queries with IN, EXISTS, correlated sub queries, queries with modified comparison operations, SELECT INTO operation, UNION operation. Sub queries in the HAVING clause</p> <p>Views: Meaning of view, Data independence provided by views, creating, altering dropping, renaming and manipulating views using SQL.</p>	6
4.	<p>Transaction Management and Concurrency Control: Concept of a transaction, ACID properties, Serial and serializable schedules, Conflict and View Serializabilty, Precedence graphs and test for conflict serializability.</p> <p>Enforcing serializability by locks: Concept of locks, the locking scheduler, Two phase Locking, upgrading and down grading locks, Concept of dead locks, Concurrency control by time stamps, The Thomas Write rule.</p> <p>Crash Recovery: ARIES algorithm, The log based recovery, recovery related structures like transaction and dirty page table, check points, recovery from a system crash.</p>	6

5.	<p>Introduction to SQL : SQL commands - Data Definition Language Commands, Data Manipulation Language Commands, The Data types a cell can hold; insertion of data into the tables; Viewing of data into the tables; Deletion operations; updating the contents of the table; modifying the structure of the table; renaming table; destroying tables; Data Constraints; Type of Data Constraint; Column Level Constraint; Table Level Constraint; Null value Concepts; The UNIQUE Constraint; The PRIMARY constraint; The FOREIGN key constraint; The CHECK Constraint; Viewing the User Constraints.</p> <p>Viewing The Data : Computations on Table Data; Arithmetic Operators; Logical Operators; Comparison Operators; Range Searching; Pattern Searching.</p> <p>Functions: Aggregate function, Date function, String functions.</p>	6
----	--	---

Reference Books

1. Ramakrishnam, Gehrke, "Database Management Systems", McGraw- Hill.
2. Ivan Bayross, "SQL,PL/SQL -The Programming language of Oracle", B.P.B. Publications
3. Elsmasri and Navathe, "Fundamentals of Database Systems", Pearson Education.

Practical:

1. Draw ER diagram for the given scenario and convert it into table.
2. Study of Data Definition Language Statement
3. Study of Data Manipulation Language Statement
4. Study of various type of JOINS.
5. Study of subqueries with all its clauses.
6. Study of various types of SET OPERATORS.
7. Study of various types of views.
8. Study of different functions.
9. Study of Transaction (Commit/ Rollback), Locks
10. Implementing deadlocks.

Case Study1

Railway Reservation

Aim: The railway reservation system facilitates the passengers to enquire about the trains available on the basis of source and destination, booking and cancellation of tickets, enquire about the status of the booked ticket, etc.

The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers. The record of a train includes its number, name, source, destination, and days on which it is available, whereas record of train status includes dates for which tickets can be booked, total number of seats available, and number of seats already booked. The database has been developed and tested on Oracle.

Description:

Passengers can book their tickets for the train in which seats are available. For this, passengers have to provide the desired train number and the date for which ticket is to be booked. Before booking a ticket for a passenger, the validity of train number and booking date is checked.

Once the train number and booking date are validated, it is checked whether the seat is available. If yes, the ticket is booked with confirmed status and corresponding ticket ID is generated which is stored along with other details of the passenger. After all the available tickets are booked, certain numbers of tickets are booked with waiting status. If the waiting lot is also finished, then tickets are not booked and a message of non-availability of seats is displayed.

The ticket once booked can be canceled at any time. For this, the passenger has to provide the ticket ID (the unique key). The ticket ID is searched and the corresponding record is deleted.

With this, the first ticket with waiting status also gets confirmed.

List of Assumption

Since the reservation system is very large in reality, it is not feasible to develop the case study to that extent and prepare documentation at that level. Therefore, a small sample case study has been created to demonstrate the working of the reservation system. To implement this sample case study, some assumptions have been made, which are as follows:

1. The number of trains has been restricted to 5.
2. The booking is open only for next seven days from the current date.
3. Only two categories of tickets can be booked, namely, *AC* and *General*.
4. The total number of tickets that can be booked in each category (*AC* and *General*) is 10.
5. The total number of tickets that can be given the status of waiting is 2.

6. The in-between stoppage stations and their bookings are not considered.

Description of Tables and Procedures

Tables and procedures that will be created are as follows:

1. **TrainList:** This table consists of details about all the available trains. The information stored in this table includes train number, train name, source, destination, fair for AC ticket, fair for general ticket, and weekdays on which train is available.

Constraint: The train number is unique.

2. **Train_Status:** This table consists of details about the dates on which tickets can be booked for a train and the status of the availability of tickets. The information stored in this table includes train number, train date, total number of AC seats, total number of general seats, number of AC seats booked, and number of general seats booked.

Constraint: Train number should exist in TrainList table.

3. **Passenger:** This table consists of details about the booked tickets. The information stored in this table includes ticket ID, train number, date for which ticket is booked, name, age, sex and address of the passenger, status of reservation (either confirmed or waiting), and category for which ticket is booked.

Constraint: Ticket ID is unique and the train number should exist in the TrainList table.

4. **Booking:** In this procedure, the train number, train date, and category is read from the passenger. On the basis of the values provided by the passenger, the corresponding record is retrieved from the Train_Status table. If the desired category is AC, then the total number of AC seats and number of booked AC seats are compared in order to find whether a ticket can be booked or not. Similarly, it can be checked for the general category. If a ticket can be booked, then passenger details are read and stored in the Passenger table.

5. **Cancel:** In this procedure, ticket ID is read from the passenger and corresponding record is searched in the Passenger table. If the record exists, it is deleted from the table. After deleting the record (if it is confirmed), the first record with waiting status for the same train and the same category are searched from the Passenger table and its status is changed to confirm.

4. WEB PROGRAMMING II
at Semester II
(Implemented during Academic Year 2022-23)
(wef 2020-21)

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	XML	6
2	PHP Basics	6
3	PHP- Working with Files	6
4	Advanced PHP and MySQL	6
5	Networking and XML Parsing	6
Total		30

Course Objectives:

By the end of the course, learners will be able to:

1. Analyze and evaluate the working of XML.
2. Apply how server-side programming works on the web.
3. Understand the working of web application with php as a server side scripting language.
4. Develop web applications using MySQL database
5. Apply the maintenance of MySQL database.

Course Outcome:

After completing this course learners will be able to:

- CO1: Design a structured approach to identify needs, interests, and functionality of a website. (Apply)
CO2: Describe POST and GET in form submission using PHP(Understand)
CO3: Design website with php sessions and cookies. (Create)
CO4: Design and develop a full-fledged website using php with MySQL database. (Create)
CO5: Apply and Analyze the working of website with Php and MySql. (Analyse)

Detailed Syllabus:

Module	Topics	No. of Lectures
1.	XML: Introduction to XML, Anatomy of an XML document, Creating XML Documents, Creating XML DTDs, XML Schemas, XSL	6
2.	PHP: Introduction of PHP, Server-side scripting. PHP BASICS: PHP syntax and variables, comments, types, constants, control structures, branching, looping, termination, functions, arrays, passing information with PHP, GET, POST	6
3.	PHP: formatting form variables, superglobal arrays, strings and string functions, regular expressions, arrays, number handling, basic PHP errors/problems, working with files and operating systems.	6
4.	Advanced PHP and MySQL: MYSQL basics, PHP/MySQL Functions, Integrating web forms and databases, authenticating your users, Displaying queries in tables.	6
5.	Building Forms from queries, String and Regular Expressions, Sessions, Cookies and HTTP, handling file uploads Networking- E-Mail, securing your website, XML parsing	6

Reference Books

1. XML: The Complete Reference –Heather Williamson, Mcgrawhill India,2001
2. Beginning php and mysql from novice to professional 4th edition 2010
3. MySQL-PHP Database Applications-Jay Greenspan and Brad Bulger,M&T Books
4. Practical PHP and MySQL, Jono Bacon, Prentice Hall,2007

Practical:

1. XML
 - a. Design a DTD, corresponding XML document and display it in browser using CSS.
 - b. Design an XML document and display it in browser using XSLT.
 - c. Design XML Schema and corresponding XML document.
2. PHP Basics-II
 - a. Write a PHP Program to accept a number from the user and print it factorial.
 - b. Write a PHP program to accept a number from the user and print whether it is prime or not.

3. PHP Basics- II
 - a. Write a PHP code to find the greater of 2 numbers. Accept the no. from the user.
 - b. Write a PHP program to display the following Binary Pyramid:
1
0 1
1 0 1
0 1 0 1.
4. String Functions and arrays
 - a. Write a PHP program to demonstrate different string functions.
 - b. Write a PHP program to create one dimensional array.
5. PHP and Database
 - a. Write a PHP code to create: (i) Create a database Student (ii) Create a table Marks (Sno,Sname,Marks)
 - b. Write a PHP program to create a database named “College”. Create a table named “Student” with following fields (sno, sname, percentage). Insert 3 records of your choice. Display the records with different suitable queries.
 - c. Design a PHP page for authenticating a user.
 - d. Write a program to send email with attachment
6. Write a program to demonstrate use of sessions and cookies.
7. Create a shopping cart using php and Mysql.
8. Write a program to demonstrate XML parsing with php.
9. Design a web page demonstrating Platform as a service (PAAS) with google cloud.
10. Demonstrate json with php.

5. DISCRETE MATHEMATICS II

at Semester II
(Implemented during Academic Year 2022-23)
(wef 2020-21)

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1	Matrices and Determinants	9
2	Linear Transformation of Matrices and Counting Principles	9
3	Permutations and Combinations	9
4	Graphs	9
5	Trees	9
	Total	45

Objective:

By the end of the course, learners will be able to:

1. Build a mathematical foundation for the computing applications
2. Lay a foundation for data structures
3. Recall matrices, properties, rank and understand its linear transformation.
4. Understand application of counting principle.
5. Define graphs and trees and their traversing.

Outcome:

After completing this course, learners will be able to:

- CO1:** Find a mathematical solution to the problems.(Apply)
CO2: Link the mathematical concepts with application in the computing domain.(Analyse)
CO3: Find Normal form of matrix, similarity and its applications.(Evaluate)
CO4: Solve problems on counting principle. (Remember)
CO5: Identify graphs and trees, their traversing and operations on binary search tree.(Understand)

Detailed Syllabus:

Module	Topics	No. of Lectures
1.	<p>Matrices: Introduction, Matrix Arithmetic, Properties of matrices, Transposes and Powers of Matrices, Inverse of a matrix, Elementary transformation, Rank of matrix, Echelon or normal form, Linear equations, Linear dependence and independence of vectors.</p> <p>Determinants: General definition, determinants and inverses of 2 x 2 matrices, Properties of determinants.</p>	9
2.	<p>Linear Transformation of matrices: Characteristics roots and characteristics vectors, their properties, Cayley-Hamilton theorem, Similarity of matrices, Reduction of matrix to a diagonal matrix.</p> <p>Counting Principles: Sum and Product Rules, Two-way counting, Tree diagram for solving counting problems, Pigeonhole Principle (without proof); Simple examples, Inclusion Exclusion Principle (Sieve formula) (Without proof).</p>	9
3.	<p>Permutations and Combinations: Partition and Distribution of objects, Permutation with distinct and indistinct objects, Binomial numbers, Combination with identities: Pascal Identity, Vandermonde's Identity, Pascal triangle, Binomial theorem, Combination with indistinct objects.</p>	9
4.	<p>Graphs: Definition and elementary results, Adjacency matrix, path matrix, Representing relations using diagraphs, Warshall's algorithm- shortest path , Linked representation of a graph, Operations on graph with algorithms – searching in a graph; Insertion in a graph, Deleting from a graph, Traversing a graph- Breadth-First search and Depth-First search</p>	9
5.	<p>Trees: Definition and elementary results. Ordered rooted tree, Binary trees, Complete and extended binary trees, representing binary trees in memory, traversing binary trees, binary search tree, Algorithms for searching and inserting, traversing binary trees, binary search tree, Algorithms for searching and inserting, in binary search trees, Algorithms for deleting in a binary search tree</p>	9

Reference Books

1. Discrete Mathematics with Applications, Sussana S. Epp, Cengage Learning, 4th Edition, 2010.
2. Discrete Mathematics, Schaum's Outline Series, Seymour Lipschutz, Tata McGraw Hill, 2007.
3. Discrete Mathematics and its Applications, Kenneth H. Rosen, Tata McGraw Hill, 2015.
4. Discrete Mathematical Structures, 6th Ed., Kolmann R. C. Busby, S. Ross, PHI, 2009.
5. Elements of Discrete Mathematics, 4th Ed., C. L. Liu, D. P. Mohapatra, Tata McGraw Hill, 2012.

6. IT platforms, Tools and Practices

*at Semester II
(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Free and Open Source Softwares	9
2	Coding Practices	9
3	Introduction to Semantic Web	9
4	The Multidisciplinary Nature of Environmental Studies	9
5	Green Computing	9
	Total	45

Course Objectives:

By the end of the course, learners will be able to:

- To prepare students according to the industry standards.
- To give an awareness on industry practices and ethics.
- To encourage the use of IT Tools so as to enable students to improve their skills and knowledge.
- To impart skills that can enable students to approach business problems analytically.
- To understand what Green IT is and How it can help improve environmental Sustainability.

Course Outcome:

After completing this course, learners will be able to:

CO1: Follow the industry standards and practices in coding.(Apply)

CO2: Illustrate various green IT services and its roles.(Understand)

CO3: Describe the importance of IT enabled services and challenges.(Understand)

CO4: Evaluate various IT tools and services for betterment of knowledge.(Evaluate)

CO5: Use and Examine different computing services.(Analyze)

Detailed Syllabus:

Module	Topics	No. of Lectures
1.	Free and Open Source Softwares Introduction: Introduction: Open Source, Free Software, Free Software vs. Open Source software, Public Domain Software, FOSS does not mean no cost. History: BSD, The Free Software Foundation and the GNU Project. GitHub: Introduction to GitHub, preparing the environment: Install Git on our machine, create a GitHub account, create a workspace on our machine GitHub workflow and the environment: Creating a repository, creating workspace, cloning the repository, creating a branch, committing the changes, merging the changes, Introduction to Wikipedia, contributing to Wikipedia	9
2.	Coding Practices: Variable Naming Conventions, Constant Naming Conventions, Indentations, General Practices, Commenting, Advantages of Coding guidelines, pair programming/code review, Refactoring, Reduction of Complexity, JavaScript best Practices, Java best Practices, Python best Practices, Python best practices (PEP 8), Code analysis tools (Pylint)	9
3.	Introduction to Semantic Web: Limitations of current Web – Development of Semantic Web – Emergence of the Social Web – Social Network analysis: Development of Social Network Analysis – Key concepts and measures in network analysis – Electronic sources for network analysis: Electronic discussion networks, Blogs and online communities – Web-based networks – Applications of Social Network Analysis.	9
4.	The Multidisciplinary Nature of Environmental Studies - Definition, Scope and Importance; Natural Resources – Introduction, Renewable and Non-renewable Resources; Ecosystems - Concept of an ecosystem, Structure and functions of an ecosystem, Producers, consumers and decomposers, Energy flow in the ecosystem, Types of ecosystem; Environmental Pollution – Definition, Causes, Effects and Control Measures; Human population and the environment.	9
5.	Green IT: Overview and Issues ; Minimizing Power Usage: Power Problems, Monitoring Power Usage, Servers, Low-Cost Options, Reducing Power Use, Low-Power Computers, PCs, Linux, Components, Servers, Computer Settings, Storage, Monitors, Power Supplies, Wireless Devices, Software; Going Paperless: Paper Problems, The Environment, Costs: Paper and Office, Practicality, Storage, Destruction, Going Paperless, Organizational Realities, Changing Over, Paperless Billing, Handheld Computers vs. the Clipboard, Unified Communications, Intranets, What to Include, Building an Intranet, Microsoft Office SharePoint Server 2007, Electronic Data Interchange (EDI), Nuts and Bolts, Value Added Networks, Advantages, Obstacles; Recycling; Green Data Storage: Introduction , Storage Media Power Characteristics, Energy Management Techniques for Hard Disks, System-Level Energy Management; Staying Green.	9

Reference Books

Unit1-

1. Open Source Initiative: <https://opensource.org/>
2. Github: <https://help.github.com/>
3. <https://medium.freecodecamp.org/how-you-can-learn-git-and-github-while-youre-learning-to-code-7a592ea287ba>
4. <https://medium.com/quick-code/top-tutorials-to-learn-git-for-beginners-622289ffdf5>
5. Wikipedia: <https://en.wikipedia.org/>

Unit 2:

1. https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide/Coding_Style
2. <https://www.castsoftware.com/glossary/coding-in-software-engineering-best-practices-good-standards>

Unit 3:

1. Peter Mika, —Social Networks and the Semantic Web, First Edition, Springer 2007.
2. Borko Furht, —Handbook of Social Network Technologies and Applications, 1st Edition, Springer, 2010.

Unit 4:

1. TextBook for Environmental Studies, Erach Bharucha.

Unit 5:

1. Toby Velte, Anthony Velte, Robert Elsenpeter, Green IT, McGraw Hill, 2008
2. Alvin Galea, Michael Schaefer, Mike Ebberts, Green Data Center: Steps for the Journey, Shroff Publishers and Distributers, 2011

Practical:

1. Create a project, repository and workspace in GitHub
2. Clone the Repositories in GitHub.
3. Creating a branch in GitHub.
4. Committing the changes in GitHub.
5. Merging the changes in GitHub.
6. Working on Wikipedia.
7. Implementing coding practices in Python using Pylint.
8. Implementing coding practices in Python using PEP8.
9. Implementing coding practices in JavaScript.
10. Green IT : Case Study

Evaluation Scheme

Test– 20 Marks

It will be conducted either as a written test or using any open source learning management system such as Moodle (Modular object-oriented dynamic learning environment) or a test based on an equivalent online course on the contents of the concerned course (subject) offered by or build using MOOC (Massive Open Online Course) platform or a written test conducted in the class room.

Assignments/Presentation: 15 marks

5 Marks - Active participation in routine class instructional deliveries:

Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

Semester End Examination- 60 Marks

Duration - 2 Hours.

Theory question paper pattern:-

All questions are compulsory.		
Question	Based on	Marks
Q1	Unit 1, 2, 3, 4, 5	4 out of 5 questions (05 marks each)
Q2	Unit 1, 2, 3, 4, 5	2 out of 3 questions (07 marks each)
Q3	Unit 1, 2, 3, 4, 5	2 out of 3 questions (08 marks each)
Q4	Based on multiple Units	1 out of 2 questions (10 marks)

Practical Examination – 50 marks (Duration: 2 Hours)

- Each practical course carries 50 Marks : 40 marks + 05 marks (journal)+ 05 marks(viva)
- Minimum 75% practical from each core/allied course are required to be completed and written in the journal.

(Certified Journal is compulsory for appearing at the time of Practical Examination)

Nagindas Khandwala College (Autonomous)

**Syllabus and Question Paper Pattern
of Courses of**

Bachelor of Science Computer Science Programme

Second Year

Semester III and IV

Under Choice Based Credit, Grading and Semester System

(Implemented during Academic Year 2022-23)

SECOND YEAR-Computer Science
(Implemented during Academic year 2022-23)

Sr. No.	Semester III	Subject code	Total Marks	Credits	Sr. No.	Semester IV	Subject code	Total marks	Credits
	Discipline Specific Compulsory Course (DSC)					Discipline Specific Compulsory Course (DSC)			
1	DSC-3 Descriptive Statistics	2231UISST	60-40 100	3	1	DSC-4 Probability Theory	2241UISPT	60-40 100	3
	Skill Enhancement Compulsory Course (SEC)					Skill Enhancement Compulsory Course (SEC)			
2	SEC-3 Core Java	2232UISCJ	60-40 100	2	2	SEC-4 Design Patterns	2242UISDP	60-40 100	2
	Core Java Practical	2232UISCJP	50	2		Design Patterns Practical	2242UISDPP	50	2
	Core Course (CC)					Core Course (CC)			
3	CC-7 Data Structures	2233UISDS	60-40 100	2	3	CC-10 Design and Analysis of Algorithms	2243UISDA	60-40 100	2
	Data Structures Practical	2233UISDSP	50	2		Design and Analysis of Algorithms Practical	2243UISDAP	50	2
4	CC-8 Computer Networks	2234UISCN	60-40 100	2	4	CC-11 System Programming	2244UISSP	60-40 100	2
	Computer Networks Practical	2234UISCNP	50	2		System Programming Practical	2244UISSPP	50	2
5	CC-9 Database Management Systems- II	2235UISDB	60-40 100	2	5	CC-12 Software Engineering	2245UISSE	60-40 100	2
	Database Management Systems- II Practical	2235UISDBP	50	2		Software Engineering Practical	2245UISSEP	50	2
	Discipline Specific Elective (DSE) CHOOSE ANY ONE					Discipline Specific Elective (DSE) CHOOSE ANY ONE			
6	Advanced Web Programming – I	2236UISAW	60-40 100	2	6	Advanced Web Programming - II	2246UISAW	60-40 100	2

Advanced Web Programming – I Practical	2236UISAWP	50	2		Advanced Web Programming – II Practical	2246UISAWP	50	2
Hybrid Mobile Application Development – I	2236UISMD	60-40 100	2		Hybrid Mobile Application Development - II	2246UISMD	60-40 100	2
Hybrid Mobile Application Development – I Practical	2236UISMDP	50	2		Hybrid Mobile Application Development – II Practical	2246UISMDP	50	2
Computer Graphics and Animation	2236UISCG	60-40 100			Advanced Java	2246UISAJ	60-40 100	2
Computer Graphics and Animation Practical	2236UISCGP	50			Advanced Java Practical	2246UISAJP	50	2
Embedded Systems	2236UISES	60-40 100	2		Multimedia Systems	2246UISMS	60-40 100	2
Embedded Systems Practical	2236UISESP	50	2		Multimedia Systems Practical	2246UISMSP	50	2
Theory of Computation - I	2236UISTC	60-40 100	2		Internet of Things	2246UISIT	60-40 100	2
Theory of Computation - I Practical	2236UISTCP	50	2		Internet of Things Practical	2246UISITP	50	2
			2		Theory of Computation - II	2246UISTC	60-40 100	2
			2		Theory of Computation - II Practical	2246UISTCP	50	2
	TOTAL		23		TOTAL			23

1. CORE JAVA

at Semester III

(Implemented during Academic Year 2022-23)

(wef 2020-21)

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Java Basics and Datatypes	6
2	Control Flow Statements, Iterations, Classes and objects	6
3	Inheritance, Packages and Multithreading	6
4	Enumerations and Arrays, I/O Streams, Exceptions handling	6
5	GUI development using AWT	6
Total		30

Course Objectives:

By the end of the course, learners will be able to:

1. Understand the importance of Object Oriented paradigm in Application development.
2. Study Java language Basics.
3. Implement Object oriented concepts using Java.
4. Understand concepts of packages and Multithreading in Java.
5. Explore the importance of Exception handling in program design.
6. To develop GUI Applications using AWT.

Course Outcome:

After successful completion of this course, learners will be able to:

- CO1: Acquire knowledge about Java language.(Understand)
- CO2: Apply Object Oriented paradigm in Application development.(Apply)
- CO3: Develop user defined packages.(Understand)
- CO4: Implement Single threaded and Multithreaded programs in Java language.(Apply)
- CO5: Create programs using Exception Handling.(Understand)
- CO6: Integrate important concepts of OOP to develop GUI applications.(Create)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	<p>Java basics: History, Architecture and its components, Java Class File, Java Runtime Environment, The Java Virtual Machine, JVM Components, The Java API, Java platform, Java development kit, Lambda Expressions, Features of Java, Java program structure, main method of Java, Command line arguments, Wrapper classes, Java Tokens, Java Statements,</p> <p>Data Types: primitive and user defined types, Arrays, Typecasting, Autoboxing and Unboxing ,Operators in Java</p>	6
2	<p>Control Flow Statements: If...Else If...Else Statement, Switch...Case Statement Iterations: While Loop, Do ... While Loop, For Loop, For each Loop, Labeled Statements, Break and Continue statements, Return Statement</p> <p>Classes and objects : Scope of variables, Defining class ,instance variables and instance methods, Creating and initializing object from class, Method Overloading, Variable Arguments [Varargs], Constructors, this instance, super instance, this instance, static members of a class, Garbage collection</p>	6
3	<p>Inheritance: Inheritance and Access Specifiers, forms of inheritance, Constructors in inheritance, super keyword, Abstract Classes and Interfaces: Abstract Classes, Abstract Methods, Interfaces, What Is An Interface? Difference between Interface and an Abstract Class, Multiple Inheritance implementation, Packages and Imports: benefits of packages, predefined packages Creating user defined Packages, Importing Packages, Using A Package.</p> <p>Multithreading: Thread concept, thread control methods, thread life cycle, the main thread, creating a thread, extending the thread class, Implementing Runnable interface.</p>	6
4	<p>Enumerations and Arrays: Creating and using enum type data, Two and Multi-Dimensional Arrays, Variable size arrays, , Creating and using Vectors, Annotations, Strings: String and String buffer class, String methods, manipulating Strings, Introduction to Collection framework.</p> <p>I/O Streams: Introduction, Byte-oriented streams, Character- oriented streams, File :Reading and writing files, Random access File, Exceptions handling: Introduction, Pre-Defined Exceptions, Try-Catch-Finally, Throws, throw, User Defined Exception examples.</p>	6
5	<p>GUI development using AWT: Introduction, Components, Event-Delegation-Model, Listeners, Layouts, Individual components Label, Button, Check Box, Radio Button, Choice, List, Menu, Text Field, Text Area</p>	6

Reference Books

1. A.Balaguruswamy, "Programming with JAVA", A Primer, TMH, 1999.
2. Core Java for Beginners by Sharanam Shah
3. Head First Java by Kathy Sierra and Bert Bates ,O Reilly ,2nd edition
4. Patrick Naughton & Herbert Schildt, JAVA 2: The Complete Reference, THM, 1999

Practical:

1. Java Data types:

- a) Write a program to accept length and breadth as command line arguments and calculate area and perimeter of rectangle.
- b) Write a program to accept integer values for a, b and c which are coefficients of quadratic equation and find the solution of quadratic equation.

2. Use of operators

- a) Demonstrate the use of various types of operators supported by Java.
- b) Write a program to reverse a string.

3. Use of Control statements and Iterators

- b) Write a program to find the smallest and largest element from an array.
- a) Write a program to count the letters, spaces, numbers and other characters of an input string.

4. Using classes and objects

- a) Design a class in java which includes instance methods and instance variables and initialize them by creating object.
- b) Demonstrate the use of constructors in java.

5. Inheritance

- a. Write a java program to implement single level inheritance.
- b. Write a java program to implement multiple inheritance.

6. Polymorphism

- a) Write a java program to implement method overloading.
- b) Write a java program to implement method overriding.

7. Packages and Multithreading

- a) Create a package, add necessary classes and import the package in java class.
- b) Write a java program to implement multithreading.

8. Arrays

- a) Write a java program for multiplying two matrices and print the product for the same.
- b) Design a class which contains instance methods to sort data in ascending and descending order stored in an array.

9. String handling and Exception handling

- a) Demonstrate the use of various methods of String and StringBuffer class to manipulate strings.
- b) Demonstrate the use of try catch and finally block to create and handle your own exception.

10. I/O streams and File handling

- a) Demonstrate the use of BufferedReader and Scanner classes for taking user input from console.
- b) Write a java program to read the student data from user and store it in the file.

11 AWT

- a) Design an AWT application that contains the interface to add student information and display the same.
- b) Design a calculator based on AWT application.

2. DATA STRUCTURES

*at Semester III
(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No	Modules	No. of Lectures
1	Introduction and Array	6
2	Linked List	6
3	Stack, Queue, Recursion	6
4	Trees and Graphs	6
5	Sorting, Searching, Tree and Hashing	6
	Total	30

Course Objectives:

By the end of the course, learners will be able to:

1. To introduce the fundamental concept of data structures.
2. To emphasize the importance of data structures in developing and implementing efficient algorithms.
3. To understand the implementation of different data structures
4. To gain knowledge with respect to complexities of different algorithms
5. To understand concept of Hashing

Course Outcomes:

After successful completion of this course, learners will be able to:

- CO1: Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms (Understand)
- CO2: Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs. (Apply)
- CO3: Demonstrate different methods for traversing trees. (Apply)
- CO4: Discuss the computational efficiency of the principal algorithms for sorting, searching, and hashing.(Understand)
- CO5: Compare and contrast the benefits of dynamic and static data structures implementations. (Analyse)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	<p>Introduction: Data and Information, Data Structure, Classification of Data Structures, Abstract Data Types, Operations on Data Structure, Algorithm, Importance of Algorithm Analysis, Complexity of an Algorithm, Asymptotic Analysis and Notations.</p> <p>Array: Introduction, One Dimensional Array, Memory Representation of One Dimensional Array, Operations, Multidimensional Arrays, Memory Representation of Two Dimensional Arrays, Sparse Matrix</p>	6
2	Introduction, Types of Linked List (one way, two way, circular, header), Operations on One way and Two way Linked List (Traversing, Searching, Insertion, Deletion, Copying, Merging), Applications of the Linked list- Representation of Polynomials, Storage of Sparse Arrays.	6
3	<p>Stacks: Stack ADT, Memory Representation, Operations, Applications - Evaluating arithmetic expressions- Conversion of Infix to postfix expression, Matching Parenthesis.</p> <p>Queue: Queue ADT, Memory Representation, Operations, Circular Queue, Priority Queue, deQueue, applications of queues.</p> <p>Recursion: Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)</p>	6
4	<p>Trees: Tree ADT, tree traversals, Binary Tree ADT, expression trees, applications of trees, binary search tree ADT, Red Black tree, AVL Trees, B-Tree, B+ Tree, Heap, Applications of heap.</p> <p>Graphs: Definition, Representation of Graph, Types of graph, Breadth-first traversal, Depth-first traversal, Application- shortest path, minimum spanning trees</p>	6
5	<p>Searching- Linear Search, Binary Search.</p> <p>Sorting - Bubble sort, Selection sort, Insertion sort, Shell sort, Radix sort.</p> <p>Hashing- Hash Functions, Separate Chaining, Open Addressing, Rehashing, Extendible Hashing.</p>	6

Reference Books

1. A Simplified Approach to Data Structures, Lalit Goyal, Vishal Goyal, Pawan Kumar, SPD 1st edition 2014.
2. Schaum's Outlines Data structure, Seymour Lipschutz TMH 2nd edition 2005.
3. Data Structure and algorithm Using Python, Rance D. Necaise, 2016 Wiley India Edition.
4. Data Structure and Algorithm in Python, Michael T. Goodrich, Robertom Tamassia, M. H. Goldwasser, 2016 Wiley India Edition.
5. Data structure – A Pseudocode Approach with C, AM Tanenbaum, Y Langsamand MJ Augustein, Prentice Hall India 2nd edition 2006.

Practical:

1. Implement the following for Array:
 - a. Write a program to store the elements in 1-D array and provide an option to perform the operations like searching, sorting, merging, reversing the elements.
 - b. Write a program to perform the Matrix addition, Multiplication and Transpose Operation.
2. Implement Linked List. Include options for insertion, deletion and search of a number, reverse the list and concatenate two linked lists
3. Implement the following for Stack:
 - a. Perform Stack operations using Array implementation.
 - b. Implement Tower of Hanoi
 - c. WAP to scan a polynomial using linked list and add two polynomial.
 - d. WAP to calculate factorial and to compute the factors of a given no. (i) using recursion, (ii) using iteration.
4. Perform Queues operations using Circular Array implementation.
5. Write a program to search an element from a list. Give user the option to perform Linear or Binary search.
6. WAP to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
7. Implement the following for Hashing:
 - a. Write a program to implement the collision technique.
 - b. Write a program to implement the concept of linear probing.
8. Write a program for inorder, postorder and preorder traversal of tree.
9. Write a program to generate the adjacency matrix.
10. Write a program for shortest path diagram.
- 11.

3. COMPUTER NETWORKS

at Semester III

(Implemented during Academic Year 2022-23)

(wef 2020-21)

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Introduction and Network Models	6
2	Physical Layer and Analog and Digital signals	6
3	Data Link Layer	6
4	Network Layer	6
5	Transport and Application Layer	6
Total		30

Course Objectives:

By the end of the course learner will be able to:

1. Understand Networking Basics.
2. Explore Hardware and Software requirements for Communication Network.
3. Understand the framework of communication networks.
4. Do layer wise study of OSI Model and TCP/IP Model.
5. Compare OSI and TCP/IP Model.
6. Implement various protocols in communication.

Course Outcome:

After successful completion of this course learner will be able to:

CO1: Define Data communication and Networking concepts.(Remember)

CO2: Acquire knowledge about common equipment, standard hardware and software requirements and communication protocols.(Understand)

CO3: Study functions of all layers in OSI Model and their requirements.(Analyze)

CO4: Understand the importance of TCP/IP Model in Communication Networks. (Understand)

CO5: Compare various protocols and their requirements in communication. (Analyze)

CO6: Simulate Communications systems using various protocols and understands its real life applications. (Apply)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	Introduction: Data communications and networking concepts, network types, network topologies, categories of networks, Internet history, standards and administration. Network models: Protocol layering, OSI model and TCP/IP model, Layered Architecture, Introduction to functions of all layers, Addressing.	6
2	Physical Layer : Data and Signals, Analog and Digital signals, Periodic and non-periodic signals, properties, Composite signals, transmission impairment , Digital and analog conversions, Transmission modes, Transmission media, Multiplexing, Frequency-Division Multiplexing, Wavelength-Division Multiplexing, Time Division Multiplexing.	6
3	Data link layer: Flow and error control, Types of Errors, Redundancy, Detection versus Correction, Error detecting methods Forward error correction and error correction by retransmission, sliding window protocols, Multiple access protocols	6
4	Network layer: Connecting devices, Switched networks, Network layer protocols, IPv4 and IPv6, addressing, Packet format, Transition strategies from IPv4 to IPv6, Unicast routing protocols - RIP, OSPF, wireless technologies, Mobile IP	6
5	Transport and Application layer : Transport layer protocols, UDP -packet format, applications ,TCP - Circular buffers, segment format, Connection management and data transfer in TCP, Introduction to SCTP, SMTP, FTP, HTTP, DNS, DHCP	6

Reference Books

1. Data Communication and Networking , Behrouz A. Forouzan Tata McGraw Hill 5th edition, 2013.
2. TCP/IP Protocol Suite Behrouz A. Forouzan Tata McGraw Hill 4th edition, 2010.
3. Computer Networks , Andrew Tanenbaum pearson, 5th edition 2013.

Practical:

1. Configuring SSH protocol for secure communication.
2. Use of ping and tracet / traceroute, ipconfig / ifconfig, route and arp utilities.
3. Configure multiple layer 2 switches where every switch is connected to multiple computers to create a network. Verify their connectivity with each other.
4. Create a wireless network of multiple devices using appropriate access point.
5. Configure IP static routing.
6. Configure IP routing using RIP.
7. Configuring Simple OSPF.
8. Configuring DHCP server and client.
9. Configuring DNS, HTTP, DHCP Server and client.
10. Use of Wireshark to scan and check the packet information of following protocols: HTTP, ICMP, TCP, SMTP, POP3

4. DATABASE MANAGEMENT SYSTEM - II

*at Semester III
(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1	Fundamentals of PL/SQL	6
2	Overview of PL/SQL Control Structures	6
3	Stored Procedures	6
4	PL/SQL Records and Collections	6
5	Exception Handling and Cursors	6
	Total	30

Course Objectives:

By the end of the course, learners will be able to:

1. Develop efficient PL/SQL programming skills
2. Understanding Oracle database.
3. Designing modular applications using packages.
4. Creating triggers to solve business challenges and enforce business rules.
5. Stepping stone for RDBMS and PL/SQL structures.

Course Outcome:

After successful completion of this course, learners will be able to:

CO1: Explain the fundamental concepts of PL/SQL. (Understand)

CO2: Develop PL/SQL queries in real-time applications. (Create)

CO3: Design modular applications using packages. (Create)

CO4: Perform PL/SQL programming using concept of Cursor Management, Error Handling, Package and Triggers (Create)

CO5: Analyze PL/SQL structures like functions, procedures, cursors and triggers for database applications. (Analyze)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	<p>Fundamentals of PL/SQL: Introduction, Overview and benefits of PL/SQL, Types of PL/SQL blocks, Simple Anonymous Block, Identifiers, types of identifiers, Declarative Section, variables, Scalar Data Types, The % Type attribute, bind variables, sequences in PL/SQL expressions, Executable statements, comment the code, deployment of SQL functions in PL/SQL, nested blocks, operators, Invoke SELECT Statements in PL/SQL, Data Manipulation in the Server using PL/SQL, Save and discard transactions.</p>	6
2	<p>Overview of PL/SQL Control Structures: Conditional Control: IF and CASE Statements, IF-THEN Statement, IF-THEN-ELSE Statement, IFTHEN-ELSIF Statement, CASE Statement, Iterative Control: LOOP and EXIT Statements, WHILE-LOOP, FOR-LOOP, Sequential Control: GOTO and NULL Statements.</p>	6
3	<p>Stored Procedures : Create a Modularized and Layered Subprogram Design, differences between Anonymous Blocks and Subprograms, Create, Call, and Remove Stored Procedures, Implement Procedures Parameters and Parameters Modes, View Procedure Information, Stored Functions and Debugging Subprograms, Create, Call, and Remove a Stored Function, advantages of using Stored Functions, Invoke User-Defined Functions in SQL Statements, Packages, advantages of Packages, components of a Package, Develop a Package, Create the Package Specification and Body, Invoke the Package constructs.</p> <p>Triggers: Concept of triggers, implementing triggers - Statement Level Triggers and Row Level Triggers, Compound Trigger to Resolve the Mutating Table Error, Comparison of Database Triggers and Stored Procedures, Create Triggers on DDL Statements.</p>	6
4	<p>PL/SQL Records: Table-based records, Cursor-based records and User-defined records.</p> <p>PL/SQL Collection: Index-by tables or Associative array, Nested table, Variable-size array or Varray.</p>	6
5	<p>Exception Handling: Concept of Exception Handling, Handle Exceptions with PL/SQL, Trap Predefined and non-predefined Oracle Server Errors, User-Defined Exceptions, Propagate Exceptions, RAISE_APPLICATION_ERROR Procedure.</p> <p>Cursors: Concept of a cursor, types of cursors: implicit cursors; explicit cursor, Cursor for loops, Cursor variables, parameterized cursors.</p>	6

Reference Books

1. Ivan Bayross, "SQL,PL/SQL -The Programming language of Oracle", 4th Ed., B.P.B. Publications, 2009.
2. Murach's Oracle SQL & PL/SQL Works with All Versions Through 11G.
3. Satish Asnani, Oracle Database 12C Hands-On Sql and Pl/Sql.
4. Oracle 11g: PL/SQL Reference Oracle Press.
5. Dr. P.S.Deshpande SQL &PL/SQL for Oracle 10g Black Book

Practical:

Practical 1: PL/SQL basics

Practical 2: Control Structures

Practical 3: Creating and working with Sequence

Practical 4: Creating Procedures, Functions and Packages

Practical 5: Creating Database Triggers.

Practical 6: Working with Collections

Practical 7: Implementing Records

Practical 8: System and User-defined Exception

Practical 9: Implicit and Explicit Cursors

Practical 10: SQL Server Performance Tuning

Case Study : Car Rental Database

A car rental company has several car rental agencies across the country. The agencies rent several classes of vehicles: compact, midsize, full size, sport utility. Each vehicle is managed by one agency. Customers rent vehicles from different agencies. When a vehicle is rented, a new rental record is added to Rental table with Dateout as the current date and DateReturned is left Null. When the vehicle is returned, the record is updated filling in the DateReturned field. A portion of the car rental database schema is as follows:

Vehicle (Vehicleid, Vehicleclassid, Agencyid)

Rental(Rentalid, Vehicleid, Customerid, Dateout, DateReturned) Customer(Customerid, CustomerName, CustomerBalance) Agency(Agencyid, AgencyLocation)

VehicleClass(VehicleClassid, VehicleClassName, RentalRate)

- a) Create sequence for generating primary key value prefixed with entity starting variable. For example, to generate Vehicleid it must be in the format "V1" and so on.
- b) Implement trigger to store the record in Rental table whenever a vehicle is rented.
- c) Create a procedure to insert, update and delete records in individual table.
- d) Create a function to display total cars rented by the customer.

5. DESCRIPTIVE STATISTICS

at Semester III

*(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1	Types, collection and scrutiny of Data	9
2	Analysis of Quantitative Data	9
3	Bivariate Data	9
4	Multivariate data	9
5	Analysis of Categorical data	9
	Total	45

Course Objective:

By the end of this course, learners will be able to:

1. To understand the basic terminologies
2. To differentiate the types of data
3. To use visualization tools and to analyze the underlying pattern in the data
4. To model the data using the suitable polynomials
5. To demonstrate the association between the variables
6. To work independently on a given data set.

Course Outcome:

After completing this course the learner will be able to:

CO1: Define the terms population, sample, univariate and multivariate data, correlation, regression and odds ratio.(Remember)

CO2: Differentiate the data into different categories.(Analyse)

CO3: Analyze a given dataset using statistical techniques.(Analyse)

CO4: Use suitable visualization tools to get a better insight into the underlying dataset.(Apply)

CO5: Device a strategy to identify the associations between the variable.(Evaluate)

CO6: Fitting lines and polynomials to model the given data.(Create)

CO7: Analyze and develop a statistical model of the data collected.(Create)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	Types Of Data: Concept of a statistical population and sample from a population; Qualitative and quantitative data, nominal and ordinal data, cross sectional and time Series data, discrete and continuous data; frequency and non-frequency data. Different types of scales – nominal, ordinal ratio and interval. Collection and Scrutiny of Data: Primary data –designing a questionnaire and a schedule; checking their consistency. Secondary data- its major sources including some government publication. Complete enumeration, controlled experiments, observational studies and sample surveys, Scrutiny of data for internal consistency and detection of errors of recording ideas of cross- validation. Presenting of Data: Construction of tables with one or more factors of classification, Diagrammatic and graphical representation of grouped data. Frequency distributions, Cumulative frequency distributions and their graphical representation, histogram, frequency polygon and gives, Stem and leaf chart. Box plot.	9
2	Analysis of Quantitative Data: Univariate data-Concepts of central tendency or location, dispersion and relative dispersion, skewness and kurtosis, and their measures including those based on quantiles and moments. Sheppard's corrections for moments for grouped data (without derivation)	9
3	Bivariate Data: Scatter diagram , Product moment correlation coefficient and its properties, Coefficient of determination, Correlation ratio, Concepts of	9

	error in regression, Principle of least squares, Fitting of linear regression and related results, Fitting of curves reducible to polynomials by transformation, Rank correlation Spearman's and Kendall's measures.	
4	Multivariate data: Multiple regression, multiple correlation and partial correlation in three variables. Their measures and related results.	9
5	Analysis of Categorical data: Consistency of categorical data, Independence and association of attributes, Various measures of association for two-way and three- way classified data, Odds ratio.	9

Reference Books

1. Bhat B.R, Srivenkatramana T and Rao Madhava K.S.; Statistics: A Beginner's Text, Vol 1, New Age International (P) Ltd.
2. Croxton F.E, Cowden D.J and Kelin S : Applied General Statistics, Prentice Hall of India.

Evaluation will be Continuous Internal Evaluation (25 Marks) and project presentation and viva-voce (75 Marks)

Discipline Specific Elective (DSE) (Any one of group A)

6A. ADVANCED WEB PROGRAMMING - I

at Semester III

(Implemented during Academic Year 2022-23)

(wef 2020-21)

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1	Object Oriented Javascript	6
2	Advanced bootstrap	6
3	AJAX	6
4	Advanced jQuery	6
5	Introduction to JSON	6
	Total	30

Course Objective:

By the end of the course, learners will be able to:

1. Recall the JavaScript, bootstrap, jquery and learn the advanced technologies.
2. Define and describe Ajax working with partial refreshes.
3. Study the concept of json to store data.
4. Learn designing with bootstrap and jQuery
5. Develop a website with the latest ajax,bootstrap and jQuery and store data in json.

Course Outcome:

After completing this course learners will be able to:

CO1: Discuss the concepts of object oriented concepts with JavaScript.(Understand)

CO2: Develop websites with bootstrap, Ajax technologies and jquery.(create)

CO3: Discuss json in web applications.(Understand)

CO4: Define and discuss major concepts, tools, techniques, and methods of web application development.(Create)

CO5: Apply the technologies learned in creation of websites.(Apply)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	Object Oriented Javascript: Javascript - Error Handling, Javascript – Validations, Javascript – Animation, Javascript – Multimedia, Javascript – Debugging, Javascript - Image Map, Javascript – Browsers, JS form validation, JS email validation, JavaScript OOPs, JS Class, JS Object, JS Prototype, JS constructor Method, JS static Method, JS Encapsulation, JS Inheritance, JS Polymorphism, JS Abstraction,JS Cookies, Cookie Attributes, Cookie with multiple Name, Deleting Cookies, JavaScript Events, JS this Keyword, JS Debugging, JS Hoisting, JS Strict Mode, JS TypedArray	6
2	Advanced bootstrap: Introduction, Grid Basic, Typography, Colors, Tables, Images, Jumbotron, Alerts, Buttons, Button Groups, Badges, Progress Bars, Spinners, Pagination, List Groups, Cards, Dropdowns, Collapse, Navs, Navbar, Forms, Inputs, Input Groups, Custom Forms, Carousel, Modal, Tooltip, Popover, Toast, Scrollspy, Utilities, Flex, Icons, Media Objects, Filters, Bootstrap 4 Grid: Grid System, Stacked/Horizontal, Grid XSmall, Grid Small, Grid Medium, Grid Large, Grid XLarge	6
3	What is AJAX? Asynchronous Ajax Technologies, AJAX - Browser Support, AJAX – Action XMLHttpRequest, How AJAX Works?, Java AJAX, AJAX with Database, Email Finder Comment Form, AJAX - Security, AJAX – Issues , <u>AJAX with PHP</u> , <u>AJAX Applications</u>	6
4	Advanced jquery jQuery Introduction, jQuery Syntax, jQuery Selectors, jQuery Events, jQuery in HTML, jQuery Get, jQuery Set, jQuery Add, jQuery Remove ,jQuery CSS Classes, jQuery css(), jQuery Dimensions, jQuery Traversing, jQuery Ancestors, jQuery Descendants, jQuery Siblings, jQuery Filtering, jQuery AJAX, Intro jQuery, Load jQuery, Get/Post, jQuery Selectors, jQuery DOM	6
5	Introduction to JSON, What is JSON, JSON - Syntax, JSON - DataTypes, JSON - Objects, JSON - Schema, JSON vs XML, JSON Parse, JSON Stringify, JSON Object, JSON Array, JSON Comments, JSON with PHP, JSON with Python, JSON with Ajax, JSON with Java	6

Reference Books

Textbooks:

1. Javascript2.0: The Complete reference, Thomas Powell and Fritz Schneider,2nd Edition
2. Learning Bootstrap: Aravind Shenoy, Ulrich Sossou, Packt Publishing,2014
3. Learning JQuery: Jonathan Chaffer, Karl Swedberg, Packt Publishing, 2013
4. Ajax for dummies, Steve Holzner

Websites:

www.w3schools.com

Practical:

1. Write a program on basic object oriented javascript.
2. Write a program to
 - a) Demonstrate Form validation with all controls
 - b) Demonstrate Encapsulation, Inheritance, polymorphism, abstraction.
3. Write a program to demonstrate cookies and events.
4. Write a program to demonstrate bootstrap components.
5. Write a program to demonstrate bootstrap grid
6. Write a program to demonstrate simple Ajax and Ajax with php
- 7 Write a program to demonstrate advanced jquery.
8. Write a program to demonstrate jquery with ajax, jquery DOM
9. Write a program to demonstrate JSON with XML.
10. Write a program to demonstrate JSON with PHP and JSON with Ajax

7A. HYBRID MOBILE APPLICATION DEVELOPMENT - I

at Semester III

(Implemented during Academic Year 2022-23)

(wef 2020-21)

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1	AngularJS Introduction	6
2	AngularJS Components	6
3	Introduction to Ionic Software setup	6
4	Ionic Components	6
5	JS Components	6
	Total	30

Course Objective:

By the end of the course, learners will be able to:

1. Focus in this course is on the basic understanding of web frameworks and
2. Develop API's for user interface design by Angular JS and Ionic Framework for Mobile Application Development.
3. On the completion of the course, students will be able to develop Hybrid mobile applications.

Course Outcome:

After completing this course learners will be able to:

CO1: Create a fully functional HTML5 app for any of the three OSES (Create)

CO2: Use PhoneGap to package HTML5 apps into native apps. (Apply)

CO3: Understand mobile application development and deployment process. (Understand)

CO4: Understand jQuery and jQuery Mobile architecture. (Understand)

CO5: Learn how to build apps with the Ionic framework. (Create)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	AngularJS Introduction, AngularJS Expressions, AngularJS Modules, AngularJS Directives, AngularJS Model, AngularJS Data Binding, AngularJS Controllers, AngularJS Scopes, AngularJS Filters, AngularJS Services, AngularJS Http, AngularJS DOM, AngularJS Events, AngularJS Forms, AngularJS Validation	6
2	AngularJS Tables, AngularJS Select, AngularJS SQL, AngularJS API, AngularJS Includes, AngularJS Animations, AngularJS Routing, AngularJS Application, AngularJS AJAX, AngularJS Views, AngularJS Scopes, AngularJS Services, AngularJS Dependency Injection, AngularJS Custom Directives, AngularJS Internationalization	6
3	Introduction to Ionic: Mobile Hybrid Architecture, What is Apache Cordova?, What is Ionic?, Features of Framework, Framework Advantages, Framework Limitation, Software setup: Install Node.js, Install Git, Install Bower, Install Gulp, Install Sublime Text, Install Cordova and Ionic CLI, The platform guide, Creating First Ionic App, The browser developer tools setup, The Ionic project structure, The config.xml file, The www folder, Scaffolding the tabs template, Scaffolding the side menu template, generator-ionic, Installing generator-ionic	6
4	Ionic Colors, Content, Header, Footer, Buttons, Lists, Cards, Forms, Toggle, Checkbox, RadioButton, Range, Select, Tabs, Grid, Icons, Padding	6
5	JS ActionSheet, JS Backdrop, JS Content, JS Form, JS Events, JS Header, JS Footer, JS Keyboard, JS List, JS Loading, JS Modal, JS Navigation, JS Popover, JS Popup, JS Scroll, JS Side Menu, JS Slide Box, JS Tabs, Cordova Integration, Admob, Camera, Facebook, In App Browser, Native Audio, GeoLocation, Media, Splashscreen	6

Reference Books

1. Pro AngularJS – Adam Freeman
2. AngularJS Programming by example - AgusKurniawan
3. AngularJS UI Development – Amit Ghart, Matthias Nehlsen.
4. Learning Ionic - Build Hybrid Mobile Applications with HTML5 - Arvind Ravulavaru
5. Ionic Framework by Example - Sani Yusuf
6. Full Stack Mobile App with Ionic Framework - Hoc Phan
7. Ionic Framework - Building Mobile Apps with Ionic Framework - Michael Bohner

Practical:

1. Data Binding- Write the Script to get the name of the User
2. Angular JS- Using Scope
 - a. Cars in an Array
 - b. Colors in List
3. Controllers - Root Scope ,Scope Colours
4. Filters - Birthday Party Invite (Use Filters) Uppercase , Lowercase. Currency, Date Friend List
5. Design a Page - Fruit Shop that has a Click Event added to an Image and MouseOver Event
6. Create a Form with Validation for Booking a Football Turf
7. Write an AngularJS program to do the following:
 1. Http
 2. Calendar Table
 3. Animation
8. Angular JS Advanced Practicals
 1. Implement Date Range Picker
 2. Create Search Filter Pipe
 3. Drag Range Slider with NGX Slider
 4. Angular 12- Star Rating
 5. Angular Radio Button
9. Mini Project on Angular JS
10. Installation of Ionic Practical
11. Ionic 2/Ionic 4-Create, Generate and Add Pages
12. Theming and Styling Ionic 2 Apps

Discipline Specific Elective (DSE) (Any one of group B)

8B. COMPUTER GRAPHICS AND ANIMATION

at Semester III

(Implemented during Academic Year 2022-23)

(wef 2020-21)

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1	Introduction and Scan Conversions Algorithms	6
2	2D and 3D Transformation	6
3	Viewing in 3D and Object Rendering	6
4	Visible-Surface Determination and Curves	6
5	Animation	6
	Total	30

Course Objectives:

By the end of the course, learners will be able to:

1. Introduce the different graphics systems and become familiar with the working of graphics system components.
2. Understand the working of different scan conversion algorithms.
3. Learn the basic principles of 2- dimensional and 3- dimensional computer graphics.
4. Transform the object using various transformation techniques.
5. Provide an understanding of mapping from world coordinates to device coordinates, clipping, and projections.
6. Understand the concept of illumination, shading and Visible Surface determination and different techniques involved in it.
7. Have a basic understanding of Animation and its principles.

Course Outcome :

After completing this course learner will be able to:

CO1: Understand different scan conversion algorithms, apply it using programming language and define their applications.(Apply)

CO2: Discuss 2D and 3D transformations and different transformation matrix used. (Understand)

CO3: Apply various 2D transformations on a 2D object. (Apply)

CO4: Discuss different shading models and Visible-Surface Determination techniques. (Understand)

CO5: Define Animations and apply the basic principles of animation. (Remember, Apply)

CO6: Create basic 2D animation using programming language. (Create)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	Introduction and Scan Conversions Algorithms Introduction to computer graphics: Introduction to computer Graphics, Computer Graphics Application, Description of some graphics devices, Hard Copy devices, display technologies, Raster-Scan Graphics Displays, CRT Raster Scan Basics, Random-Scan Display Processor, LCD displays. Scan conversion – Digital Differential Analyser (DDA) algorithm, Bresenham's Line drawing algorithm. Bresenham's method of Circle drawing, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Clipping Lines algorithms: Cohen-Sutherland, Clipping Polygons.	6
2	2D and 3D Transformation 2D Transformations: Basic transformation, Matrix representation and Homogeneous coordinates, Rotation, Reflection, Scaling, Transformation of Points, Transformation of The Unit Square, Combined transformation, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line. 3D Transformations: translation, rotation, scaling, composite-shears and reflections, Three-dimensional viewing, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations, Projection and its types.	6
3	Viewing in 3D and Object Rendering Stages in 3D viewing, Canonical View Volume (CVV), specifying an Arbitrary 3D View, Examples of 3D Viewing, the Mathematics of Planar Geometric Projections, Combined transformation matrices for projections and viewing, Coordinate Systems and matrices, camera model and viewing pyramid. Object Rendering: Introduction Object-Rendering, Light Modelling Techniques, Illumination Model, Shading, Flat Shading, Polygon Mesh Shading, Gouraud Shading Model, Phong Shading, Transparency Effect,	6

	Shadows, Texture and Object Representation, Ray Tracing, Ray Casting, Radiosity, Color Models.	
4	<p>Visible-Surface Determination and Curves Visible-Surface Determination: Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, The z-Buffer Algorithm, Scan-line method, Painter’s algorithms (depth sorting), Area subdivision method, BSP trees, Visible-Surface Ray Tracing, comparison of the methods.</p> <p>Plane Curves and Surfaces: Curve Representation, Nonparametric Curves, Parametric Curves, Parametric Representation of a Circle, Ellipse, Parabola, Hyperbola, Representation of Space Curves, Cubic Splines, Bezier Curves, B-spline Curves, Parametric Cubic Curves, Quadric Surfaces. Bezier Surfaces</p>	6
5	<p>Animation Principles of Animation, about motion graphics, Principles of animation, Key framing, Deformations, Character Animation, Physics-Based Animation, Procedural Techniques, Groups of Objects, Animation by computer, Animation file formats, Display of animated content, Introduction to Morphing, Three-Dimensional Morphing, Motion Tweening and Motion Editor, Classic tweening and Shape tweening.</p>	6

Reference Books

1. Computer Graphics, R. K. Maurya, John Wiley.
2. Mathematical elements of Computer Graphics, David F. Rogers, J. Alan Adams, TataMcGraw-Hill.
3. Procedural eteinem« of Computer Graphics, David F. Rogers, Tata McGraw-Hill.
4. Multimedia BASICS by Suzanne (Suzanne Weixel) Weixel, Jennifer Fulton, Karl Barksdale and CherylBeck Morse (Mar 14, 2003)

Practical:

1. Solve the following:
 - a) Draw a coordinate axis at the center of the screen.
 - b) Divide your screen into four regions, draw a circle, rectangle, ellipse and half ellipse in each region with appropriate message.
2. Draw a simple hut on the screen.
3. Write a program to implement the DDA line drawing algorithm.
4. Write a program to implement Bresenham’s line drawing algorithm.
5. Write a program to implement a midpoint circle drawing algorithm.
6. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.
7. Write a program to clip a polygon using Sutherland Hodgeman algorithm.

8. Write a program to apply various 2D transformations on a 2D object (use homogeneous coordinates).
9. Write a program to demonstrate 2D animation such as clock simulation or rising sun.
10. Write a program to implement the bouncing ball inside a defined rectangular window.

9B. EMBEDDED SYSTEMS

*at Semester III
(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1	Introduction and Core Components	6
2	Embedded Hardware	6
3	Introduction to 8051 Microcontroller	6
4	Programming in C	6
5	RTOS and Trends	6
	Total	30

Course Objectives:

By the end of the course, learners will be able to:

1. Understand the meaning, components of a basic embedded systems
2. Study the characteristics and quality attribute of an embedded systems
3. Understand the memory structure of embedded systems and its peripheral devices
4. Differentiate aspects of programming for developing embedded systems
5. Understand the EDLC

Course Outcome:

After completion of this course, the learner should be able to:

CO1: Define and describe the characteristics of an embedded system. (Remember)

CO2: Explain the internal architecture and interfacing of different peripheral devices with Microcontrollers. (Understand)

CO3: Identify the need and role of embedded systems in industry. (Understand)

CO4: Write the programs for 8051 microcontroller using C/Assembly Programming language. (Apply)

CO5: Breakdown the pinout connection of 8051 Microcontroller in different categories. (Analyse)

CO6: Demonstrate the usage of Embedded Systems in real life applications such as traffic signals, elevators, and so on. (Apply)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	<p>Introduction: Embedded Systems and general purpose computer systems, history, classifications, applications and purpose of embedded systems</p> <p>Core of embedded systems: microprocessors and microcontrollers, RISC and CISC controllers, Big endian and Little endian processors, Application specific ICs, Programmable logic devices, COTS, sensors and actuators, communication interface, embedded firmware, other system components.</p>	6
2	<p>Characteristics and quality attributes of embedded systems: Characteristics, operational and non-operational quality attributes.</p> <p>Embedded Hardware: Memory map, i/o map, interrupt map, processor family, external peripherals, memory – RAM, ROM, types of RAM and ROM, memory testing, CRC ,Flash memory.</p> <p>Peripherals: Control and Status Registers, Device Driver, Timer Driver - Watchdog Timers.</p>	6
3	<p>The 8051 Microcontrollers: Microcontrollers and Embedded processors, Overview of 8051 family.8051 Microcontroller hardware, Input/output pins, Ports, and Circuits, External Memory.</p> <p>Designing Embedded System with 8051 Microcontroller: Factors to be considered in selecting a controller, why 8051 Microcontroller, Designing with 8051.</p>	6
4	<p>8051 Programming in C: Variables, Declaring a variable, Data Types and time delay in 8051 C, I/O Programming, Logic operations, Data conversion Programs, structure of embedded program, infinite loop, compiling, linking and debugging, functions.</p>	6
5	<p>Real Time Operating System (RTOS): Operating system basics, types of operating systems, Real-Time Characteristics, Selection Process of an RTOS.</p> <p>Design and Development: Embedded system development Environment – IDE, types of file generated on cross compilation, disassembler/ decompiler, simulator, emulator and debugging, embedded product development life-cycle, trends in embedded industry.</p>	6

Reference Books

1. Shibu K.V, —Introduction to Embedded Systems, McGraw Hill.2014

2. Jonathan W.Valvano, —Embedded Microcomputer Systems Real Time Interfacing, Third Edition Cengage Learning, 2012
3. Raj Kamal, —Embedded Systems-Architecture, Programming and Design, 3 edition, TMH.2015
4. Lyla, —Embedded Systems, Pearson , 2013
5. David E. Simon, —An Embedded Software Primer, Pearson Education, 2000.

Practical:

1. A. To interface 8 LEDs at Input-output port and create different patterns.
B. To demonstrate timer working in timer mode and blink LED without using any loop delay routine.
2. Configure timer control registers of 8051 and develop a program to generate given time delay.
3. To demonstrate use of general purpose port i.e. Input/ output port of two controllers for data transfer between them.
4. Serial I / O: Configure 8051 serial port for asynchronous serial communication with serial port of PC exchange text messages to PC and display on PC screen.
5. To demonstrate interfacing of seven-segment LED displays and generate counting from 0 to 9 with fixed time delay.
6. A. Interface 8051 with D/A converter and generate a square wave of given frequency on an oscilloscope.
B. Interface 8051 with a D/A converter and generate triangular waves of given frequency on the oscilloscope.
C. Using a D/A converter generates sine wave on oscilloscope with the help of lookup table stored in data area of 8051.
7. Interface stepper motor with 8051 and write a program to move the motor through a given angle in clock wise or counter clock wise direction
8. Generate traffic signal
9. Implement Temperature controller
10. Implement Elevator control

10B. THEORY OF COMPUTATION I

at Semester III

(Implemented during Academic Year 2022-23)

(wef 2020-21)

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1	Mathematical Theory and Algebraic Structures	6
2	Introduction and General Concepts of Automata Theory	6
3	Regular Expressions and Finite Automata	6
4	Regular Languages and Finite Automata	6
5	Pumping Lemma for regular Languages	6
	Total	30

Course Objectives:

The theoretical foundations of computer science have expanded substantially in recent years. The objective of this course is to:

- To give an overview of the theoretical foundations of computer science from the perspective of formal languages
- To illustrate finite state machines to solve problems in computing
- To explain the hierarchy of problems arising in the computer sciences.
- To construct automata for any given pattern and find its equivalent regular expressions

Course Outcome:

After completing this course the learner will be able to:

CO1: Express computer science problems as mathematical statements and to formulate proofs. (Understand)

CO2: Develop a machine model for the computing problem (Create)

CO3: Use basic concepts of finite automata techniques (Apply)

CO4: Represent a pattern as regular expression (Understand)

CO5: Design Finite Automata for different Regular Expressions and Languages(Create)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	<p>Review of Mathematical Theory: Sets, Functions, Logical statements, Propositional Logic, Propositional equivalences, Predicates and Quantifiers, Proofs, relations, languages, Mathematical induction, strong principle, Recursive definitions</p> <p>Algebraic Structures: Properties, Semi Group, Groups, Subgroups, Abelian Group, Cyclic Groups, Lagrange's Theorem, Normal subgroups, Rings, Euclidean ring, Fields, Introduction to Vector Space – Basis , Dimension.</p>	6
2	<p>Introduction to automata theory: Introduction to regular expressions, Finite Automata, Structural Representations, Finite state automata – Deterministic Finite Automata (DFA), Nondeterministic Finite Automata (NFA), Finite Automata with ϵ- Transitions, Eliminating ϵ-transitions, Equivalence of NFA and DFA, Conversion of NFA to DFA, Finite automata with output (Moore and Mealy machines)</p> <p>General Concepts of Automata Theory: Alphabets Strings, Languages, Applications of Automata Theory, Formal representation of languages – Chomsky Classification.</p>	6
3	<p>Regular Expression (RE): Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA's to Regular Expressions, Converting Regular Expressions to Automata, applications of Regular Expressions.</p> <p>Regular Expressions and Finite Automata: The Operators of regular Expressions, Building Regular Expressions, Precedence of Regular-Expression Operators, Precedence of Regular-Expression Operators Finite Automata and Regular Expressions: From DFA's to Regular Expressions, Converting DFA's to Regular Expressions by Eliminating States.</p>	6
4	<p>Regular Languages and Finite Automata: Regular grammars and FA, FA for regular grammar, Regular grammar for FA.</p> <p>Converting Regular Expressions to Automata: Converting Regular Expressions to NFA, Myhill-Nerode Theorem, Equivalence of Deterministic and Nondeterministic Finite Automata, Minimization of Deterministic Finite Automata</p>	6

5	Properties of Regular Languages: A Criterion for Regularity, Pumping Lemma for Regular Languages, Applications of the Pumping Lemma Closure Properties of Regular Languages, Decision Properties of Regular Languages.	6
---	---	---

Reference Books

1. Kenneth H. Rosen, "Discrete Mathematics And Its Applications", 7th Ed, McGrawHill, 2012.
2. Erwin Kreyszig, "Advanced Engineering Mathematics", Wiley India, 9th Edition 2011.
2. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, "Introduction to Automata Theory, Languages, and Computation", Second Edition, Prentice-Hall, 2001
3. John Hopcroft, Rajeev Motwani, and Jeffrey Ullman, Automata Theory, Languages, and Computation

Additional Reading:

1. An introduction to automata theory and formal languages By Adesh K. Pandey, Publisher: S.K. Kataria & Sons
2. Introduction to computer theory By Deniel I. Cohen , Joh Wiley & Sons, Inc
4. Compiler Design By Alfred V Aho, Addison Wesley
5. Introduction to the Theory of Computation By Michael Sipser
6. Peter Dehning, Jack B. Dennis, "Machines, Languages and Computation", Second Edition, Prentice-Hall, 1978
7. Harry R. Lewis, Christos H. Papadimitriou, "Elements of the theory of computation", Second Edition, Prentice-Hall, 1998

Practical:

- Preparation of power-point presentation, which include videos, animations, pictures, graphics for better understanding theory and practical work. Faculty in-charge will allocate chapters/ parts of chapters to groups of students so that the entire syllabus to be covered. The power-point slides should be put up on the LMS, along with the names of the students of the group, the name of the faculty, Department and College on the first slide.

OR

- Mini-project based on the implementation aspects of the theory topics covered. Faculty in-charge will allocate chapters/ parts of chapters to groups of students so that the entire syllabus to be covered.

Evaluation Scheme

Test– 20 Marks

It will be conducted either as a written test or using any open source learning management system

such as Moodle (Modular object-oriented dynamic learning environment) or a test based on an equivalent online course on the contents of the concerned course (subject) offered by or build using MOOC (Massive Open Online Course) platform or a written test conducted in the class room.

Assignments/Presentation: 15 marks

5 Marks - Active participation in routine class instructional deliveries:

Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

Semester End Examination- 60 Marks

Duration - 2 Hours.

Theory question paper pattern:-

All questions are compulsory.		
Question	Based on	Marks
Q1	Unit 1, 2, 3, 4, 5	4 out of 5 questions (05 marks each)
Q2	Unit 1, 2, 3, 4, 5	2 out of 3 questions (07 marks each)
Q3	Unit 1, 2, 3, 4, 5	2 out of 3 questions (08 marks each)
Q4	Based on multiple Units	1 out of 2 questions (10 marks)

Practical Examination – 50 marks (Duration: 2 Hours)

- Each practical course carries 50 Marks : 40 marks + 05 marks (journal)+ 05 marks(viva)
- Minimum 75% practical from each core/allied course are required to be completed and written in the journal.

(Certified Journal is compulsory for appearing at the time of Practical Examination)

***Syllabus of Courses of
Bachelor of Science in Information Technology Programme
at Semester IV
(Implemented during Academic Year 2022-23)***

1. DESIGN PATTERNS
at Semester IV
(Implemented during Academic Year 2022-23)
(wef 2020-21)

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Introduction	6
2	Designing a Document Editor	6
3	Creational Patterns	6
4	Structural Pattern	6
5	Behavioral Patterns	6
Total		30

Course Objectives:

By the end of the course, learners will be able to:

1. Understand design patterns.
2. Acquire basic understanding of commonly used to design patterns to solve problems.
3. Compare the object-oriented programming model with the standard structured programming.
4. Uses the basic design principles in solving real life problems.
5. Understand the necessity of dealing with change.
6. Learn to apply the pattern based analysis and design to the software to be developed.

Course Outcome:

After completing this course, learners will be able to:

CO1: Learn the role of design patterns in software development.(Understand)

CO2: Understands the pattern based design principle.(Understand)

CO3: Apply a fundamental set of design patterns utilizing object-oriented principles to solve real-world software design problems.(Apply)

CO4: Able to work individually as well as in teams to create reusable and cohesive software components.(Apply)

CO5: Create Design Patterns to solve real world problems.(Create)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	What is a design pattern? design patterns in MVC, Describing Design Patterns, The Catalog of Design Patterns, Organizing the Catalog, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.	6
2	A Case Study: Designing a Document Editor: Design Problems, Document Structure, Formatting, Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations Spelling Checking and Hyphenation, Summary.	6
3	Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton, Discussion of Creational Patterns.	6
4	Structural Pattern Part-I : Adapter, Bridge, Composite. Structural Pattern Part-II : Decorator, façade, Flyweight, Proxy.	6
5	Behavioral Patterns Part-I : Chain of Responsibility, Command, Interpreter, Iterator. Behavioral Patterns Part-II : Mediator, Memento, Observer, State, Strategy, Template Method ,Visitor, Discussion of Behavioral Patterns.	6

Reference Books

1. Design Patterns By Erich Gamma, Pearson Education.
2. Head First Design Patterns By Eric Freeman-O'reilly-spd.

Practical:

Practical 1: Implement Abstract Factory and Factory Method patterns

Practical 2: Implement Builder and Singleton patterns

Practical 3: Implement Prototype pattern

Practical 4: Implement Adapter and Bridge patterns

Practical 5: Implement Composite and Decorator patterns

Practical 6: Implement Chain of Responsibility pattern

Practical 7: Implement Iterator pattern

Practical 8: Implement Template pattern

Practical 9: Implement Memento pattern

Practical 10. Implement Facade pattern

2. DESIGN AND ANALYSIS OF ALGORITHMS

*at Semester IV
(Implemented during Academic Year 2022-23)*

(wef 2020-21)

Modules at a Glance

Sr. No	Modules	No. of Lectures
1	Fundamentals of Algorithmic Problem Solving	6
2	Applications	6
3	Computing a Binomial Coefficient	6
4	The Simplex Method	6
5	Limitations of Algorithm Power	6
	Total	30

Course Objective:

By the end of the course, learner will be able to:

1. Understand algorithms
2. Design efficient solutions for real-world problems.
3. Analyze and compare various algorithms
4. Understand and analyse the problems solvable in polynomial time and non-deterministic polynomial time.

Course Outcome:

After the completion of the course the learners will be able to:

CO1: Analyze the asymptotic performance of algorithms (Analyze)

CO2: Write rigorous correctness proofs for algorithms. (Apply)

CO3: Demonstrate familiarity with major algorithms and data structures and explain the NP completeness. (Understand)

CO4: Apply important algorithmic design paradigms and methods of analysis. (Apply)

CO5: Synthesize efficient algorithms in common design situations. (Analyze)

CO6: Develop suitable algorithms for a given problem. (Create)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	The notion of an Algorithm – Fundamentals of Algorithmic Problem Solving – Important Problem Types – Fundamentals of the Analysis of Algorithm Efficiency – Analysis Framework – Asymptotic Notations and its properties – Mathematical analysis for Recursive and Non-recursive algorithms.	6
2	Brute Force – Closest-Pair and Convex-Hull Problems-Exhaustive Search – Traveling Salesman Problem – Knapsack Problem – Assignment problem. Divide and conquer methodology – Merge sort – Quick sort – Binary search – Multiplication of Large Integers – Strassen’s Matrix Multiplication-Closest-Pair and Convex-Hull Problems.	6
3	Computing a Binomial Coefficient – Warshall’s and Floyd’s algorithm – Optimal Binary Search Trees – Knapsack Problem and Memory functions. Greedy Technique– Prim’s algorithm- Kruskal’s Algorithm- Dijkstra’s Algorithm-Huffman Trees.	6
4	The Simplex Method-The Maximum-Flow Problem – Maximum Matching in Bipartite Graphs- The Stable marriage Problem.	6
5	Limitations of Algorithm Power-Lower-Bound Arguments-Decision Trees-P, NP and NP-Complete Problems–Coping with the Limitations – Backtracking – n-Queens problem – Hamiltonian Circuit Problem – Subset Sum Problem-Branch and Bound – Assignment problem – Knapsack Problem – Traveling Salesman Problem- Approximation Algorithms for NP-Hard Problems – Traveling Salesman problem – Knapsack problem.	6

Reference Books

1. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, “Introduction to Algorithms”, Third Edition, PHI Learning Private Limited, 2012.
2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, “Data Structures and Algorithms”, Pearson Education, Reprint 2006.
3. Donald E. Knuth, “The Art of Computer Programming”, Volumes 1& 3 Pearson Education, 2009. Steven S. Skiena, “The Algorithm Design Manual”, Second Edition, Springer, 2008.

Practical:

1. Sort a given set of elements using different sorting methods and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
2. Implement Knapsack problem using Dynamic Programming
3. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
4. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.
5. Perform various tree traversal algorithms for a given tree.
6. Print all the nodes reachable from a given starting node in a digraph using both BFS and dfs methods.
7. Implement any scheme to find the optimal solution for the Traveling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.
8. Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.
9. Implement N Queen's problem using Backtracking.
10. Find the longest palindromic subsequence using dynamic programming.

3. SYSTEM PROGRAMMING

at Semester IV

(Implemented during Academic Year 2022-23)

(wef 2020-21)

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Linux Utilities	6
2	Files and Directories	6
3	Process and signal	6
4	Inter Process Communications, message queues, semaphores	6
5	Shared Memory and sockets	6
Total		30

Course Objectives:

By the end of the course, learners will be able to:

1. Understand the basics of OS concepts, efficient scripts and utilities are to be used.
2. Learn the concept of files and directories.
3. Describe the working of processes and signals.
4. Describe the concept of IPC, semaphores, memory and sockets.
5. Design and implement code generators using C and gdb

Course Outcome:

After the completion of the course, the students would be able to

CO1: Understand and make effective use of Linux utilities and Shell scripting language (bash) to solve problems.(Apply)

CO2: Develop the skills necessary for systems programming including file system programming, process and signal management and inter-process communication.(Analyse)

CO3: Develop the basic skills required to write network programs using sockets.(Apply)

CO4: Design and implement system utility programs.(Create)

CO5: Describe UNIX file systems and process control.(Understand)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	<p>Linux Utilities: File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text Processing utilities and backup utilities.</p> <p>sed- scripts, operation, address, commands, applications, awk- execution, fields and records, scripts, operations, patterns, actions, associative arrays, string and mathematical functions, system commands in awk, applications.</p> <p>Shell programming with the Bourne again shell(bash): Introduction, shell responsibilities, pipes and Redirection, Here documents, Running a shell scripts, The shell as a programming language, Shell meta characters, File name substitution, Shell variables, Command substitution, Shell commands, The environment, Quoting, Test command, control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts</p>	6
2	<p>Files and Directories: File Concept, File types ,File system Structure, File meta data – Inodes, Kernel support for files, System calls for I/O operations – open, create, read, write, lseek, dup2. File status information – stat family, file and record locking, fcntl function, Links – Soft links & hard links – symlink, link, unlink.</p> <p>Directories – creating, removing, changing directories – mkdir, rmdir, chdir, obtaining current working directory – getcwd, directory contents, scanning directories – opendir, readdir, closedir, rewinddir functions.</p>	6
3	<p>Process: Process concepts, layout of C program image in main memory, process environment –environment list, environment variables, getenv, setenv, Kernel support for process, process identification, process control-process creation. Replacing a process image, Waiting for a process, process termination, zombie process, orphan process, system call interface form process management – fork, vfork, exit, wait, waitpid, exec family, process groups, session and controlling terminal, difference between threads and processes.</p> <p>Signal- Introduction to signals, Signal generation and handling, Kernel support for signal, Signal function, unreliable signals, reliable signals, Kill, raise, alarm, pause, abort, sleep functions.</p>	6
4	<p>Inter Process Communications:- Introduction to IPC, IPC between processes on a single computer, IPC between processes on different systems, pipes – creating, IPC between related processes using Unnamed Pipes, FIFOs – creation, IPC between unrelated processes using FIFO (named pipes),</p>	6

	<p>difference between named and unnamed pipes, popen and pclose library functions.</p> <p>Message Queues – kernel support for messages, APIs for Message Queues, client/server examples.</p> <p>Semaphores – Kernel support for semaphores, APIs for semaphores, FILE locking with semaphores.</p>	
5	<p>Shared Memory:- Kernel support for Shared memory, APIs for shared memory, shared memory examples.</p> <p>Sockets:- Introduction to Berkeley Sockets, IPC over a network, client/server model, Sockets Address Structure(UNIX Domain & Internet Domain), Socket System calls for connection oriented Protocol and connectionless protocol, Example client/server programs – single server-client connection, multiple simultaneous clients, socket options – setsockopt and fcntl system calls, comparison of IPC mechanisms.</p>	6

Reference Books

1. Unix concepts and Applications, 4th Edition, Sumitabha Das, TMH.
2. Beginning Linux Programming, 4th Edition, N.Matthew, R.Stones, Wrox, Willey India Edition.

Practical:

1. Installation of Unix/Linux operating system
2. Basic Commands
3. Write a shell script program to display list of user currently logged in
4. Shell script program to check whether given file is a directory or not.
5. Write a program using sed command to print duplicated lines of Input.
6. Write a grep/egrep script to find the number of character, words and lines in a file.
7. Write an awk script to develop a Fibonacci series.
8. Write a shell script to change the priority of processes.
9. Write a shell script program to check variable attributes of file and processes.
10. To execute programs using gdb to utilize its various features like breakpoints, conditional breakpoints etc.
11. Program to run multiple child processes using fork().

4. SOFTWARE ENGINEERING

*at Semester IV
(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1	Introduction to Software Engineering and SDLC	6
2	Introduction to Software Requirements Specifications	6
3	System Design and Object-oriented design using UML	6
4	Software Measurement and Metrics and Testing	6
5	Project management and Software Maintenance and Risk Analysis	6
	Total	30

Course Objective:

By the end of the course, learners will be able to:

1. Understand the basic theory of software engineering.
2. Understand the software development life cycle.
3. Understand and apply the basic project management practices in real life projects.
4. Understanding of approaches to verification and validation including static analysis, and reviews.
5. Describe software measurement and software risks.

Course Outcome:

After completing this course, learners will be able to:

CO1: Decompose the given project in various phases of a lifecycle. (Analyse)

CO2: Choose appropriate process model depending on the user requirements. (Evaluate)

CO3: Perform various life cycle activities like Analysis, Design, Implementation, Testing and Maintenance.(Apply)

CO4: Know various processes used in all the phases of the product. (Understand)

CO5: Apply the knowledge, techniques, and skills in the development of a software product. (Apply)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	<p>Introduction to Software Engineering: Introduction to Software, Types of Software, Classes of Software, Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Myths, Software Applications, Software-Engineering Processes, Programs Versus Software Products.</p> <p>Software-Development Life-Cycle Models Software-Development Life-Cycle, Waterfall Model, Prototyping Model, Spiral Model, Evolutionary Development Model, Iterative-Enhancement Model, RAD Model, Agile Model.</p>	6
2	<p>Introduction to Software Requirements Specifications Requirement Engineering, Process of Requirements Engineering, Information Modeling, Data-Flow Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS Documents, SRS Validation, Components of SRS, Characteristics of SRS.</p> <p>Software Reliability and Quality Assurance Verification and Validation, Software Quality Assurance, Software Quality, International Standard Organization (ISO), Comparison of ISO-9000 Certification and the SEI-CMM, Reliability Issues, Reliability Metrics, Reliability Growth Modeling, Reliability Assessment.</p>	6
3	<p>System Design: System/Software Design, Architectural Design, Low-Level Design Coupling and Cohesion, Functional-Oriented Versus, object-Oriented Approach, Design Specifications, Verification for Design, Monitoring and Control for Design.</p> <p>Object-oriented design using UML - Class diagram, Object diagram, Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram, Component diagram, Deployment diagram.</p>	6
4	<p>Software Measurement and Metrics: Software Metrics, Halstead's Software Science, Function-Point Based Measures, Cyclomatic Complexity</p> <p>Software Testing: Introduction to Testing, Testing Principles, Testing Objectives, Test Oracles, Levels of Testing, White-Box Testing/Structural Testing, Functional/Black-Box Testing, Test Plan, Test Case Design.</p>	6
5	<p>Project management: Revision of Project Management Process, Role of Project Manager, Project Management Knowledge Areas, Managing Changes in requirements, Software-Project Estimation, Constructive Cost Model (COCOMO).</p>	6

	Software Maintenance and Risk Analysis: Software as an Evolution Entity, Software-Configuration Management Activities, Change-Control Process, Software-Version Control, Software-Configuration Management, Need for Maintenance, Categories of Maintenance, Maintenance Costs, Software-Risk Analysis and Management.	
--	--	--

Reference Books

1. R.S. Pressman, Software Engineering: A Practitioner's Approach (7th Edition), McGrawHill, 2009.
2. P. Jalote, An Integrated Approach to Software Engineering (2nd Edition), Narosa Publishing House, 2003.
3. K.K. Aggarwal and Y. Singh, Software Engineering (revised 2nd Edition), New Age International Publishers, 2008.
4. I. Sommerville, Software Engineering (8th edition), Addison Wesley, 2006.
5. Software Engineering & Testing , B.B.Agarwal, S.P.Tayal, M.Gupta.

Practical:

1. Problem Statement on Case Study/Mini Project
2. ER Diagram
3. Use Case Diagram
4. Class Diagram
5. Activity Diagram
6. Sequence Diagram
7. State Machine Diagram
8. Component Diagram
9. Project Management
 - Computing FP
 - Gantt chart
 - Estimation Effort and schedule
10. Creating Test Cases using following testing types
 - Black Box Testing
 - White Box Testing

5. PROBABILITY THEORY
at Semester IV
(Implemented during Academic Year 2022-23)
(wef 2020-21)

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1	Important Concept in Probability	9
2	Random variables	9
3	Standard univariate discrete distributions	9
4	Continuous univariate distributions	9
5	Chebyshev's inequality and applications	9
	Total	45

Course Objective:

By the end of the course, learners will be able to:

1. Understand the fundamentals of probability theory
2. Explain the random variable and the underlying distribution
3. Apply the chebychev's inequality and central limit theorem

Course Outcome:

After the completion of the course the learners will be able to:

- CO1: Analyze a given dataset using statistical techniques. (Analyse)
- CO2: Demonstrate the probability distributions (Understand)
- CO3: Model the given data using a suitable distribution (Apply)
- CO4: Demonstrate the properties of the data in terms of the distribution. (Analyse)
- CO5: Apply chebychev's inequality and central limit theorem (Apply)
- CO6: work on a real data set and make a statistical model for analysis and prediction (Create)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	Important Concept in Probability: Definition of probability – classical and relative frequency approach to probability, Richard Von Mises, Cramer and Kolmogorov’s approaches to probability, merits and demerits of these approaches (only general ideas to be given) Random Experiment : Trial, sample point and sample space, definition of an event, Operation of events, mutually exclusive and exhaustive events. Discrete sample space, properties of probability based on axiomatic approach , conditional probability independence of events, Baye’s theorem and its applications.	9
2	Random variables: Definition of discrete random variables, probability mass function , idea of continuous random variable, probability density function , illustrations of random variables and its properties, expectation of a random variable and its properties moments, measures of location, dispersion, skewness and kurtosis, probability generating function (if it exists), their properties and uses.	9
3	Standard univariate discrete distributions and their properties, Discrete uniform, Binomial, Poisson , Hypergeometric, and Negative Binomial distributions .	9
4	Continuous univariate distributions – uniform, normal, Cauchy, Laplace , Exponential, ChiSquare, Gamma and Beta distributions ,Bivariate distribution (including marginal and conditional distributions).	9
5	Chebyshev’s inequality and applications, statements and application of weak law of large numbers and central limit theorems. De moivre’s central limit theorem, Lindberg Levy form of central limit.	9

Reference Books

1. Bhat B.R, Srivenkatramana T and Rao Madhava K.S.; Statistics: A Beginner’s Text, Vol 1, New Age International (P) Ltd.
2. Edward P.J. Ford J.S. and Lin : Probability for Statistical Decision – Making , Prentice Hall.

Evaluation will be Continuous Internal Evaluation (25 Marks) and project presentation and viva-voce (75 Marks)

DISCIPLINE SPECIFIC ELECTIVE (Any ONE from GROUP A)

6A. ADVANCED WEB PROGRAMMING - II

*at Semester IV
(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	AngularJS Introduction	6
2	AngularJS Components	6
3	Node.js Introduction	6
4	Node.js Components	6
5	Node.js with MongoDB	6
Total		30

Pre-requisite: Student must have completed ADVANCED WEB PROGRAMMING – I course in Semester III.

Course Objectives:

By the end of the course, learners will be able to:

1. Understand and learn Angular JS concepts and develop web applications with its components.
2. Understand and learn Node JS environment and develop web applications with MongoDB database.
3. Explain and implement the components of AngularJS.
4. Develop web applications with Node JS
5. Implement MongoDB with Node JS

Course Outcome:

After completing this course learners will be able to:

- CO1: Understand the concept of Angular JS (Understand)
- CO2: Describe the working of Angular JS with its components.(Understand)
- CO3: Recognize the concept of Node JS usage in web application.(Analyse)
- CO4: Ability to develop web application with Angular JS and Node JS.(Apply)
- CO5: Create and develop node JS applications with MongoDB.(Create)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	AngularJS Introduction, AngularJS - MVC Architecture, AngularJS Expressions, AngularJS Modules, AngularJS Directives, AngularJS Model, AngularJS Data Binding, AngularJS Controllers, AngularJS Scopes, AngularJS Filters, AngularJS Services, AngularJS Http, AngularJS DOM, AngularJS Events, AngularJS Forms, AngularJS Validation	6
2	AngularJS Tables, AngularJS Select, AngularJS SQL, AngularJS API, AngularJS Includes, AngularJS Animations, AngularJS Routing, AngularJS Application, AngularJS AJAX, AngularJS Views, AngularJS Scopes, AngularJS Services, AngularJS Dependency Injection, AngularJS Custom Directives, AngularJS Internationalization	6
3	Node.js Introduction, Node.js - Introduction, Node.js - Environment Setup, Node.js Modules, Node.js HTTP Module, Node.js File System, Node.js URL Module, Node.js REPL Terminal, Node.js Package Manager (NPM), Node.js Callbacks Concept, Node.js NPM, Node.js Event Loop, Node.js Event Emitter	6
4	Node.js Buffers, Node.js Streams, Node.js File System, Node.js Global Objects, Node.js Utility Modules, Node.js Upload Files, Node.js Email, Node.js Web Module, Node.js Express Framework, Node.js RESTful API, Node.js Scaling Application, Node.js Packaging	6
5	Node.js with MongoDB: MongoDB introduction, MongoDB Create Database, MongoDB Create Collection, MongoDB Insert, MongoDB Find, MongoDB Query, MongoDB Sort, MongoDB Delete, MongoDB Drop Collection, MongoDB Update, MongoDB Limit, MongoDB Join	6

Reference Books

1. Pro AngularJS – Adam Freeman, APress, 2014
2. AngularJS Programming by example - AgusKurniawan, PE Press
3. AngularJS UI Development – Amit Ghart, Matthias Nehlsen, Packt Publishing
4. Beginning Node.js – Basarat Ali Syed, APress, 2014
5. Node.js Web Development – David Herron, Packt Publishing, 2018
6. Getting MEAN with Mongo, Express, Angular and Node – Simon Holmes, Manning Publications, 2018.

Practical:

1. Write a program on basic Angular JS
2. Write a program to demonstrate AngularJS form and validation.
3. Write a program to demonstrate AngularJS Ajax and AngularJS views
4. Write a program to demonstrate Node.js Package Manager.
5. Write a program to demonstrate Node.js Event Emitter.
6. Write a program to demonstrate Node.js Buffer
- 7 Write a program to demonstrate Node.js email
8. Write a program to demonstrate Node.js with json
9. Write a program to demonstrate Node.js with MongoDB with its operations.
10. Write a program to demonstrate Node.js with MongoDB Limit and MongoDB Join

7A. HYBRID MOBILE APPLICATION DEVELOPMENT - II

*at Semester IV
(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Introduction to React	6
2	Templating using JSX	6
3	Working with state and Props	6
4	React Components	6
5	React Modules	6
	Total	30

Pre-requisite: Student must have completed HYBRID MOBILE APPLICATION DEVELOPMENT - I course in Semester III.

Course Objective:

By the end of the course learner will be able to:

1. Focus in this course is on the basic understanding of web frameworks and API's for user interface design by Angular JS and Ionic Framework for Mobile Application Development.
2. On the completion of the course, students will be able to develop Hybrid mobile applications.

Course Outcome:

After completing this course learners will be able to:

CO1: Learn how to build single page applications with React JS (Understand)

CO2: Use and Install React-Native dependencies for MAC and Windows Run Android and IOS simulator (Apply)

CO3: Understand and Learn the key concepts of the NodeJS (Understand)

CO4: Understand Nodejs, learn rapidly growing web server technology, Nodejs & understand how NodeJS works with Node course! (Understand)

CO5: Learn how to Style with React-Native and flex-box rules (Analyse)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	Introduction to React: What is React?, Why React?, React version history, React 16 vs React 15, Just React - Hello World, Using create-react-app, Anatomy of react project, Running the app, Debugging first react app	6
2	Templating using JSX: Working with React. createElement, Expressions, Using logical operators, Specifying attributes, Specifying children, Fragments, Component: Significance of component architecture, Types of components, Functional, Class based, Pure, Component Composition	6
3	Working with state and Props: What is state and its significance, Read state and set state, Passing data to component using props, Validating props using propTypes, Supplying default values to props using default-Props	6
4	Rendering Lists, Event Handling, Understanding component lifecycle and handling errors. Working with Forms, Context, Code Splitting, Hooks, Routing with react route, Redux, Immutable.js, React Redux, Redux Middleware	6
5	Unit Testing, Webpack Primer, Isomorphic React	6

Reference Books

1. Learning React: Functional Web Development with React and Redux - Book by Alex Banks and Eve Porcello
2. The Road to Learn React: Your Journey to Master Plain Yet Pragmatic React. Js - Book by Robin Wieruch
3. ReactJS by Example - Building Modern Web Applications with React - Book by Prathamesh Sonpatki

Practical:

1. Install create-react-app and create a new react project.
2. Create a Todo List with different JSX expression, apply CSS via className and styles.
3. Create multi-step form in React JS
4. Create Fragments with Flexbox
5. Create a Card Component for the Author written with Multiple Books
6. Create a shopping cart using different props types.
7. Create drop down menu using react bootstrap.
8. Create a functional component that uses the ability of state and life cycle features
 1. Create counter increment and decrement.
 2. Create song playlist
9. Create a conditional rendering for a shoe store (logging in) / subscribing to the email and map function for discount.
10. Create a component lifecycle with football points and chats
11. Create a program to handle error boundaries
12. Create a library Subscription Form using maps and navigation
13. Create an Online Bidding or auction website with routing rules and components (Code Splitting and Lazy loading)
14. To create a project using Synthetic Events.

7A. ADVANCED JAVA

*at Semester IV
(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Networking and Servlets	6
2	JDBC and Cookies	6
3	JSP and MVC	6
4	Enterprise JavaBeans	6
5	Java Persistence API and Web Services	6
Total		30

Prerequisite:

To study Advanced Java basic knowledge of core java is needed.

Course Objective:

By the end of the course, learners will be able to:

- Learn to perform socket programming in java.
- Get an understanding on Enterprise Java and the servlet technology.
- Explain the database connection using JDBC.
- Understand the concept of cookies and session tracking in java.
- Work with JSP, EJB and implement it.

Course Outcome:

After completing this course learner will be able to:

- CO1: Develop networking concept using Socket Programming. (Create)
CO2: Understand Enterprise Application and Java EE architecture. (Understand)
CO3: Explain the concept of servlet, JDBC and apply it through coding. (Understand, Apply)
CO4: Learn and analyse the concept of cookies and session tracking in Java. (Analyse)
CO5: Create applications using servlet, JSP, EJB along with implementation of database. (Create)
CO6: Basic understanding of JavaBean, Web services and their applications. (Understand)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	<p>Networking (Socket Programming): The java.net package, Connection oriented transmission – Stream Socket Class. Creating a Socket to a remote host on a port (creating TCP client and server) Simple Socket Program Example.</p> <p>Introduction to Java EE: What is an Enterprise Application? What is java enterprise edition? Java EE Technologies, Java EE evolution, Glassfish server.</p> <p>Java EE Architecture, Server and Containers: Types of System Architecture, Java EE Server, Java EE Containers.</p> <p>Server side programming with Java Servlet: Introduction, The need for Dynamic content, Java Servlet Technology, Lifecycle of a Servlet, a simple welcome servlet, Servlet API, Handling get and post request (HTTP), Handling data from HTML to servlet, Retrieving data from database to servlet(Servlet-JDBC).</p>	6
2	<p>Database Programming with JDBC:</p> <p>Introduction, JDBC Architecture, Types of Drivers, Basic JDBC program Concept, Making database Connection, Standard Statement, PreparedStatement, Callable statement ResultSet, Executing SQL commands, Executing queries, Metadata and Transaction.</p> <p>Cookies: Kinds Of Cookies, Where Cookies Are Used? Creating Cookies Using Servlet, Dynamically Changing The Colors Of A Page.</p> <p>Session tracking: Introduction, Lifecycle of HTTP Session, Methods of session tracking- User Authorization, URL rewriting, Hidden form fields.</p>	6
3	<p>JSP: Why use Java Server Pages? JSP v\s Servlets, Life Cycle of a JSP Page, How does a JSP function? How does JSP execute? Simple first JSP program, JSP architecture.Implicit Objects,Scripting elements – Declarations, Expressions, and Scriplets, Comments, JSP Directives, Action elements- forward, include, use bean, get and set property. JSP with database.</p> <p>Java Beans: Introduction, Components of JavaBean, JavaBeans Properties. Examples.</p> <p>MVC: What is MVC? History of MVC, Features of MVC, MVC architecture, Components of MVC, Examples- A simple implementation of MVC using Java, Advantages and Disadvantages.</p>	6

4	<p>Introduction To Enterprise JavaBeans: What is EJB? Benefits of EJB, When to use EJB? Enterprise bean architecture, types of enterprise bean.</p> <p>Working With Session Beans: Introduction, When to use Session Beans? Types of Session Beans, Remote and Local Interfaces, Accessing Interfaces, Lifecycle of Enterprise Beans, Packaging Enterprise Beans, Example of Stateful Session Bean, Example of Stateless Session Bean, Example of Singleton Session Beans.</p> <p>Working with Message Driven Beans: Lifecycle of a Message Driven Bean, Uses of Message Driven Beans, The Message Driven Beans Example.</p> <p>Java Naming and Directory Interface: Introduction to naming and directory services, What is JNDI? Basic lookup.</p>	6
5	<p>Persistence, Object/Relational Mapping And JPA: What is Persistence? Persistence in Java, Object/Relational Mapping,</p> <p>Introduction to Java Persistence API: The Java Persistence API, JPA, ORM, Database and Application, Architecture of JPA, How JPA Works and its specification.</p> <p>Introduction to Hibernate: What is Hibernate? Why Hibernate? Hibernate, Database and the application, Component of hibernate application, Architecture of hibernate Writing Hibernate Application: Creating Hibernate Configuration File, Adding a Mapping Class, Creating JSPs, Running the hibernate Application</p> <p>Web services: Introduction, Web service components, SOAP, RESTful, SOAP v/s REST, Java Web services API. Building a web services using JAX-WS.</p>	6

References Books:

1. The Complete Reference Java- Herbert Schildt 7th edition, McGraw Hill.
2. Java EE 7 For Beginners :Sharanam Shah, Vaishali Shah : SPD : First : 2017
3. Java EE 8 Cookbook : Build reliable applications with the most or robust and mature technology for enterprise development : Elder Moraes : Packt : First : 2018
4. Advanced Java Programming :Uttam Kumar Roy : Oxford Press : 2015
5. Core Java for Beginners by Sharanam Shah

Practical:

1. Implement a program on Socket Programming.
2. Implement the following Simple Servlet applications.
 - a. Create a simple calculator application using servlet.

- b. Create a servlet for a login page. If the username and password are correct then it says message “Hello <username>” else a message “login failed”.
- c. Create a registration servlet in Java using JDBC. Accept the details such as Username, Password, Email, and Country from the user using HTML Form and store the registration details in the database.

3. Implement the following Servlet applications.

- a. Design database for student administration. Develop servlet(s) to perform CRUD operations. The program should do the following:-
 - i. Insert details for a new student.
 - ii. View all the details of the students.
 - iii. Update details of existing student.
 - iv. Delete details of a student.
- b. Develop Simple Servlet Question Answer Application using Database.

4. Programs on cookies and session in servlet.

- a. Create a servlet that uses Cookies to store the number of times a user has visited servlet.
- b. Create a servlet demonstrating the use of session creation and destruction. Also check whether the user has visited this page first time or has visited earlier also using sessions.

5. Demonstrate following JSP programs.

- a. Create Customer table in CUST database. Perform select, insert and delete operations on Customer table using JSP.
- b. Create a registration and login JSP application to register and authenticate the user based on username and password using JDBC.
- c. Develop a simple JSP application to display values obtained from the use of intrinsic objects of various types.

6. Implement following JSP applications

- a. Develop a simple JSP application to pass values from one page to another with validations. (Name-txt, age-txt, hobbies-checkbox, email-txt, gender-radio button).
- b. Create an html page with fields, eno, name, age, desg, salary. Now on submit this data to a JSP page which will update the employee table of database with matching eno.

7. Javabean program

- a. Write a Student class with three properties. The useBean action declares a JavaBean for use in a JSP. Write Java application to access JavaBeans Properties.

8. Demonstrate following EJB applications.

- a. Create a Currency Converter application using EJB.
- b. Develop a Simple Room Reservation System Application Using EJB.
- c. Develop simple shopping cart application using EJB [Stateful Session Bean].

9. Implement the following EJB applications with different types of Beans.

- a. Develop simple EJB application to demonstrate Servlet Hit count using Singleton Session Beans.
- b. Develop simple visitor Statistics application using Message Driven Bean [Stateless Session Bean].
- c. Develop simple Marks Entry Application to demonstrate accessing Database using EJB.

10. Implement following Hibernate applications.

- a. Develop a Hibernate application to store Feedback of Website Visitor in MySQL Database.
- b. Develop a Hibernate application to store and retrieve employee details in MySQL Database.

11. Programs on web services.

- a. Develop a simple “Hello World” Web Service with SOAP in Java.
- b. Develop a Simple Web Service and Client with JAX-WS.

DISCIPLINE SPECIFIC ELECTIVE (Any ONE from GROUP B)

8B. MULTIMEDIA SYSTEMS at Semester IV (Implemented during Academic Year 2022-23) (wef 2020-21)

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Introduction, Hardware and Software	6
2	Compression and Multimedia Objects	6
3	Audio and Video Compression	6
4	Multimedia with Internet and Animation	6
5	Multimedia-looking towards Future	6
Total		30

Pre-requisite: Students must have completed COMPUTER GRAPHICS AND ANIMATION course in Semester III.

Course Objectives:

By the end of the course, learners will be able to:

1. Become multimedia/graphics designers and engineers in their areas of expertise.
2. Understand the basic components of multimedia and different compression techniques used.
3. Get an understanding of Animation and Virtual Reality.
4. Implement different techniques for creating animated videos and edit different images using softwares.
5. Apply different effects and color coding on various objects.

Course Outcome:

After completing this course learner will be able to:

CO1: Use different compression techniques of text, audio, video and apply basics of animation.

(Apply)

CO2: Understand different file formats used for text, image, audio and video and compare between them. (Understand, Analyze)

CO3: Apply different animation on character, object, etc. Apply text effects, color variations on objects. (Apply)

CO4: Use different software for animation purposes and create a small animation clip and enhance graphics images using different software's. (Create)

CO5: Create different logos, cards and websites using multimedia software. (Create)

CO6: Discuss the concept of Virtual reality and its applications. (Understand)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	<p>Introduction, Hardware and Software Introduction to Multimedia: Definition, History of Multimedia, characteristics of multimedia, multimedia building blocks/components, Multimedia elements, Multimedia architecture, analog and digital representations, evolving technologies for multimedia, Multimedia applications.</p> <p>Multimedia-Hardware and Software: Multimedia Hardware – Macintosh and Windows production, Platforms, Hardware peripherals – Connections, Memory and storage devices, Media software – Basic tools, making instant multimedia, Multimedia software and Authoring tools, Production Standards.</p>	6
2	<p>Compression and Multimedia Objects Compression and coding: Definitions, need of data compression, types of compression, basic compression techniques- run length, Huffman's coding. Text, Graphics and Image data representation: Text, using text in multimedia, text file formats(txt.doc, rtf, pdf), text compression techniques. Basic Image fundamentals, image File formats - (BMP, TIFF, JPEG, GIF), image acquisition, graphics/image data types, color models in images, image processing and software, Image enhancement, image compression techniques.</p>	6
3	<p>Audio and Video Compression Audio and Audio compression: Introduction, Acoustics, Sound Waves, Types and Properties of Sounds, Psycho-Acoustics, Components of an Audio Systems, Digitization of audio, synthesizers, MIDI, digital audio processing, Quantization and Transmission of audio, Audio File Formats, audio compression techniques, audio processing software. Video and video compression: Introduction, types of video signal, television system, video color spaces, digital video, digital video processing, video file formats, video compression techniques, video recording and storage formats, video processing software</p>	6
4	<p>Multimedia with Internet and Animation: Multimedia and the Internet: History, Internet working, Connections, Internet Services, The World Wide Web, Tools for the WWW – Web Servers, Web Browsers, Web page makers and editors, Plug-Ins and Delivery Vehicles, HTML, VRML, Designing for the WWW – Working on the Web, Multimedia</p>	6

	Applications – Media Communication, Media Consumption, Media Entertainment, Media games. Animation: Basics of animation, types of animation, principles of animation, use of animation, traditional animation, computer based animation, OpenGL overview	
5	Multimedia-looking towards Future Multimedia-looking towards Future: Digital Communication and New Media, Interactive Television, Digital Broadcasting, Digital Radio, Multimedia Conferencing, Assembling and delivering a project-planning and costing, Designing and Producing, content and talent, Delivering, CD-ROM technology. Virtual Reality: Concept, Forms of VR, VR applications, VR devices: Hand Gloves, Head mounted tracking system, VR chair, VCR, 3D Sound system, Head mounted display.	6

Reference Books

1. Ranjan Parekh, "Principles of Multimedia", 2/E, Tata McGraw-Hill, ISBN: 1259006506
2. Ze-Nian Li, Marks S. Drew, "Fundamentals of Multimedia", Pearson Education
3. Keyes, "Multimedia Handbook", TMH, 2000.
4. R. Steinmetz and K. Nahrstedt, 2001, Multimedia: Computing, Communications & Applications, Pearson, Delhi.
5. S. Rimmer, 2000, Advanced Multimedia Programming, PHI, New Delhi..

Practical:

(Any open source/proprietary animation software (1-5), Photoshop (6-8), Dreamviewer (9-10))

1. To Move an object, to move an object in the path
2. Text flip, Text color change,
3. Creating a link using texts and objects, change the color of the object.
4. Shape Tweening and Using shape hints, Motion tweening, hybrid tweening.
5. Character Animation, Object Animation, Drawing Images
6. To create a greeting card, Create background picture
7. Text and Photo effects, editing images.
8. Designing Logos
9. Creating menu bar
10. Creating Pages and sites

Case Study: Create a project/ case study to demonstrate a small animation clip / create greeting card, logo or business card.

9B. INTERNET OF THINGS

*at Semester IV
(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	SoC and Raspberry Pi.	6
2	ARM 8 Architecture and Programming Raspberry Pi	6
3	Programing interfaces	6
4	Useful Implementations and Introduction to IoT	6
5	IoT Security and IoT Service as a Platform	6
Total		30

Prerequisite:

Students must have completed the EMBEDDED SYSTEMS course in Semester III.

Course Objectives:

By the end of the course, learners will be able to:

1. Understand the concept of System on Chip technology
2. Understand and implement basic concepts of Raspberry Pi
3. Understand the concept of Internet of Things
4. To configure the Raspberry Pi with the help of Linux Commands

Course Outcome:

After completion of this course, the learner should be able to:

CO1: Explain the System on Chip along with its structure and few products. (Level: Understand)

CO2: Explain the basic components of Raspberry Pi. (Level: Understand)

CO3: Write different commands of Linux programming for configuration of Raspberry Pi. (Level: Apply)

CO4: Describe the different communication interface used by Raspberry Pi. (Level: Understand)

CO5: Designing IoT based Project. (Level: Create)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	SoC and Raspberry Pi. System on Chip: What is System on chip? Structure of System on Chip. SoC products: FPGA, GPU, APU, Compute Units.	6
2	ARM 8 Architecture: SoC on ARM 8. ARM 8 Architecture Introduction Introduction to Raspberry Pi: Introduction to Raspberry Pi, Raspberry Pi Hardware, Preparing your raspberry Pi. Raspberry Pi Boot: Learn how this small SoC boots without BIOS. Configuring boot sequences and hardware. Programming Raspberry Pi Raspberry Pi and Linux: About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands	6
3	Programing interfaces: Introduction to Node.js, Python, NodeRed Raspberry Pi Interfaces: UART, GPIO, I2C, SPI	6
4	Useful Implementations: Cross Compilation, Pulse Width Modulation, SPI for Camera. Introduction to IoT: What is IoT? IoT examples, Simple IoT LED Program. IoT and Protocols	6
5	IoT Security: HTTP, UPnp, CoAP, MQTT, XMPP. IoT Service as a Platform: Clayster, Thinger.io, SenseIoT, carriots and Node RED. IoT Security and Interoperability: Risks, Modes of Attacks, Tools for Security and Interoperability.	6

Reference Books

1) Learning Internet of Things, Peter Waher, Packt Publishing(2015) 2) Mastering the Raspberry Pi, Warren Gay, Apress(2014)

Additional Reference(s):

1) Abusing the Internet of Things, Nitesh Dhanjani, O'Reilly

Practical:

1. Preparing Raspberry Pi: Hardware preparation and Installation
2. Linux Commands: Exploring the Raspbian
3. GPIO: Light the LED with Python
4. GPIO: LED Grid Module: Program the 8X8 Grid with Different Formulas
5. SPI: Camera Connection and capturing Images using SPI
6. Real Time Clock display using PWM.
7. Stepper Motor Control: PWM to manage stepper motor speed.
8. Node RED: Connect LED to Internet of Things
9. Stack of Raspberry Pi for better Computing and analysis
10. Create a simple Web server using Raspberry Pi

10B. THEORY OF COMPUTATION II

*at Semester IV
(Implemented during Academic Year 2022-23)
(wef 2020-21)*

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Context Free Grammar (CFG)	6
2	CFG and PDA	6
3	PDA and language of PDA	6
4	LBA and Turing Machine	6
5	Turing Machine and Halting Problem	6
Total		30

Pre-requisite: Student must have completed THEORY OF COMPUTATION I course in Semester III.

Course Objectives:

By the end of the course learner will be able to:

1. To familiarize Regular grammars, context free grammar
2. To design a context free grammar for any given language
3. To give an overview of the theoretical foundations of computer science from the perspective of formal languages
4. To understand Turing machines and their capability
5. To understand undecidable problems and NP class problems
6. To participate in GATE, PGECET and other competitive examinations

Course Outcome:

On completion of the course learner will be able to:

CO1: Design Finite Automata's for different Regular Expressions and Languages (apply)

CO2: Construct context free grammar for various languages (create)

CO3: Solve various problems of applying normal form techniques, push down automata and Turing Machines (apply)

CO4: Apply this basic knowledge of Theory of Computation in the computer field to solve computational problems and in the field of compiler also. (apply)

CO5: Demonstrate the working of Turing machines and their capability (understand)

CO6: To explain undecidable problems and NP class problems (understand)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	Introduction to Grammar: The Languages of a Grammar, and types of Grammars, Regular Grammar, Conversion of FA to grammar. Context-Free Grammars(CFG): Definition of Context-Free Grammars, Derivation Trees (Parse trees), Constructing Parse Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings.	6
2	Applications of Context-Free Grammars: Parsers, Ambiguity in Grammars and Languages: Ambiguous Grammars, Removing Ambiguity From Grammars, Leftmost Derivations as a Way to Express Ambiguity, Inherent Ambiguity. Minimization of CFG's, Chomsky Normal form and Greibach Normal form. Language acceptability by PDA through empty stack and final state	6
3	Pushdown Automata: Definition Formal Definition of Pushdown Automata, A Graphical Notation for PDA's, Instantaneous Descriptions of a PDA Languages of PDA: Acceptance by Final State, Acceptance by Empty Stack, From Empty Stack to Final State, From Final State to Empty Stack Equivalence of PDA's and CFG's: From Grammars to Pushdown Automata, From PDA's to Grammars	6
4	Linear Bounded Automata and Turing Machines: Formal definition and behavior, Context sensitive language and linear bounded automata (LBA) Turing Machine (TM) – Basics and formal definition, Transition Diagrams for Turing Machines, TM as a language acceptor, Turing Machines and Computers, TM as a computer of integer functions, Types of TMs.	6

5	<p>Turing Machines and Halting Programming Techniques for Turing Machines, Extensions to the Basic Turing Machine, Restricted Turing Machines, Universal Turing machine, The Halting problem.</p> <p>Recursively Enumerable Languages, Recursive languages, Properties of Recursively Enumerable Languages and Recursive Languages, Decidability and Halting Problem.</p> <p>Un-decidability: Undecidable problems about TMs, Decidability, Post's correspondence problem (PCP), undecidability of PCP.</p>	6
---	---	---

Reference Books

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2007), Introduction to Automata Theory Languages and Computation, 3rd edition, Pearson Education, India.
2. L. P Mishra, N. Chandrashekar (2003), Theory of Computer Science-Automata Languages and Computation, 2nd edition, Prentice Hall of India, India.
3. An introduction to automata theory and formal languages By Adesh K. Pandey, Publisher: S.K. Kataria & Sons
4. Michael Sipser, Introduction to the Theory of Computation, Thomson Course Technology, 2006

Practical:

- Preparation of power-point presentation, which include videos, animations, pictures, graphics for better understanding theory and practical work. Faculty in-charge will allocate chapters/ parts of chapters to groups of students so that the entire syllabus to be covered. The power-point slides should be put up on the LMS, along with the names of the students of the group, the name of the faculty, Department and College on the first slide.

OR

- Mini-project based on the implementation aspects of the theory topics covered. Faculty in-charge will allocate chapters/ parts of chapters to groups of students so that the entire syllabus to be covered

Evaluation Scheme

Test– 20 Marks

It will be conducted either as a written test or using any open source learning management system such as Moodle (Modular object-oriented dynamic learning environment) or a test based on an equivalent online course on the contents of the concerned course (subject) offered by or build using MOOC (Massive Open Online Course) platform or a written test conducted in the class room.

Assignments/Presentation: 15 marks

5 Marks - Active participation in routine class instructional deliveries:

Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

Semester End Examination- 60 Marks

Duration - 2 Hours.

Theory question paper pattern:-

All questions are compulsory.		
Question	Based on	Marks
Q1	Unit 1, 2, 3, 4, 5	4 out of 5 questions (05 marks each)
Q2	Unit 1, 2, 3, 4, 5	2 out of 3 questions (07 marks each)
Q3	Unit 1, 2, 3, 4, 5	2 out of 3 questions (08 marks each)
Q4	Based on multiple Units	1 out of 2 questions (10 marks)

Practical Examination – 50 marks (Duration: 2 Hours)

- Each practical course carries 50 Marks : 40 marks + 05 marks (journal)+ 05 marks(viva)
- Minimum 75% practical from each core/allied course are required to be completed and written in the journal.

(Certified Journal is compulsory for appearing at the time of Practical Examination)

Nagindas Khandwala College (Autonomous)

**Syllabus and Question Paper Pattern
of Courses of**

Bachelor of Science Computer Science Programme

Third Year

Semester V and VI

Under Choice Based Credit, Grading and Semester System

(Implemented from Academic Year 2022-2023)

THIRD YEAR-Computer Science
(Implemented during Academic Year 2022-2023)

Sr No	Semester V	Subject code	Total Marks	Credits	Sr. No.	Semester VI	Subject code	Total Marks	Credits
	Core Course (CC)					Core Course (CC)			
1	CC 13: Data and Network Security	2251UCSDN	60-40 100	3	1	CC 16: Building blocks of Artificial Intelligence	2261UCSBA	60-40 100	3
	Data and Network Security Practical	2251UCSPR	50	1		CC 16: Building blocks of Artificial Intelligence Practical	2261UCSPR	50	1
2	CC 14: Optimization Techniques	2252UCSOT	60-40 100	3	2	CC 17: Data Science	2262UCSDS	60-40 100	3
	CC 14: Optimization Techniques Practical	2252UCSPR	50	1		CC 17: Data Science Practical	2262UCSPR	50	1
	Discipline Specific Elective (DSE)					Discipline Specific Elective (DSE)			
	Discipline Specific Elective (DSE-5) (Any ONE)					Discipline Specific Elective (DSE-7) (Any ONE)			
3	Web Services	2253UCSWS	60-40 100	3	3	Cloud Computing	2263UCSCC	60-40 100	3
	Web Services Practical	2253UCWSPR	50	1		Cloud Computing Practical	2263UCSCCPR	50	1
4	Ethical Hacking	2253UCSEH	60-40 100	3	4	Cyber Forensics	2263UCSCF	60-40 100	3

	Ethical Hacking Practical	2253UCSEHPR	50	1		Cyber Forensics Practical	2263UCSCF PR	50	1
5	Introduction to Big Data Analytics	2253UCSBD	60-40 100	3					
	Introduction to Big Data Analytics Practical	2253UCSBDPR	50	1					
	Discipline Specific Elective (DSE-6) (Any ONE)					Discipline Specific Elective (DSE-8) (Any ONE)			
6	Wireless Sensor Networks	2254UCSWN	60-40 100	3	5	Mobile and Pervasive Computing	2264UCSMP	60-40 100	3
	Wireless Sensor Networks Practical	2254UCSWNPR	50	1	11	Mobile and Pervasive Computing Practical	2264UCSMPPR	50	1
7	Game Programming	2254UCSGP	60-40 100	3	6	Digital Image Processing	2264UCSDI	60-40 100	3
	Game Programming Practical	2254UCSGPPR	50	1	12	Digital Image Processing Practical	2264UCSDIPR	50	1
8	Web Technologies	2254UCSWT	60-40 100	3					
	Web Technologies Practical	2254UCSWTPR	50	1					
17	Core 15: Project Implementation	2255UCSPI	100	4	13	Core 18: Project Implementation	2265UCSPI	100	4
	TOTAL			20		TOTAL			20

Data and Network security
at Semester V
(Implemented during Academic Year 2022-23)

Modules at a Glance

Sr. No.	Modules	No. of lectures
1	Understanding OSI Security Architecture.	9
2	Understanding classical and Modern Encryption Techniques.	9
3	Key Management	9
4	Web Security considerations and Controlling Network Access	9
5	Transaction Intrusion	9
	Total	45

Course Objectives:

By the end of the course, learners will be able to:

- Provide conceptual understanding of network security issues, challenges and mechanisms.
- Develop basic skills of secure network architecture and understand different cryptographic algorithms.
- Describe common network vulnerabilities and attacks and understand the importance of Web security.
- Study Transport Layer Security and Intrusion Detection and prevention techniques.

Course Outcome:

After completing this course learners will be able to:

CO1: Create a basic understanding of networking domain. (Create)

CO2: Understands OSI security Architecture. (Understand)

CO3: Implement various Cryptographic Algorithms. (Apply)

CO4: Implement various protocols for Key Management as well as provide application-level security
(Apply)

CO5: Apply different Intrusion Detection and Prevention Techniques to secure Data and Network
(Apply)

Detailed Syllabus:

Module	Topics	No of Lectures
1	Introduction: Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms Classical Encryption Techniques: Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Steganography, Block Cipher Principles, The Data Encryption Standard, The Strength of DES, AES (round details not expected), Multiple Encryption and Triple DES, Block Cipher Modes of Operation, Stream Ciphers	9
2	Public-Key Cryptography and RSA: Principles of Public-Key Cryptosystems, The RSA Algorithm. Key Management: Public-Key Cryptosystems, Key Management, Diffie-Hellman Key Exchange Message Authentication and Hash Functions: Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs, Secure Hash Algorithm	9
3	HMAC Digital Signatures and Authentication: Digital Signatures, Authentication Protocols, Digital Signature Standard Authentication Applications: Kerberos, X.509 Authentication, Public-Key Infrastructure	9
4	Electronic Mail Security: Pretty Good Privacy, S/MIME IP Security: Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management Web Security: Web Security Considerations, Network access control and Cloud security,	9
5	Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction Intrusion: Intruders, Intrusion Techniques, Intrusion Detection Malicious Software: Viruses and Related Threats, Virus Countermeasures, DDOS Firewalls: Firewall Design Principles, Types of Firewalls	9

Reference Books

1. William Stallings (2014), Cryptography and Network Security: Principles and Practice 6th Edition, Pearson.
2. Atul Kahate (2013), Cryptography and Network Security, Tata McGraw-Hill
3. Behrouz A Fourouzan, Debdeep Mukhopadhyay (2011), Cryptography and Network, Second Edition, TMH.

Practical:

1. Write programs to implement the following Substitution Cipher Techniques: Caesar Cipher and Monoalphabetic Cipher
2. Write programs to implement the following Substitution Cipher Techniques: Vernam Cipher and Playfair Cipher
3. Write programs to implement the following Transposition Cipher Techniques: Rail Fence Cipher and Simple Columnar Technique
4. Write program to encrypt and decrypt strings using DES Algorithm and AES Algorithm
5. Write a program to implement RSA algorithm to perform encryption / decryption of a given string.
6. Write a program to implement the Diffie-Hellman Key Agreement algorithm to generate symmetric keys.
7. Write a program to implement the MD5 algorithm to compute the message digest.
8. Write a program to calculate HMAC-SHA1 Signature
9. Write a program to implement SSL.
10. Configure Windows Firewall to block: A port, An Program, A website

Optimization Techniques

*at Semester V
(Implemented during Academic Year 2022-23)*

Modules at a Glance

Sr. No.	Modules	No. of Lectures
---------	---------	-----------------

1	Introduction and applications of optimization.	9
2	Essential features of optimization problems	9
3	Network Analysis	9
4	Computational Complexity	9
5	Genetic Algorithms and Simulated Annealing	9
	Total	45

Course Objectives:

After completion of this program learners will be:

- Able to understand the basics of optimization techniques
- Get familiar with the Linear programming and meta-heuristic search techniques.
- Learn classical optimization techniques and numerical methods of optimization.
- Know the basics of different evolutionary algorithms.
- Explain Integer programming techniques and apply different optimization techniques to solve various models arising from engineering areas.

Course Outcome:

After completing this course learners will be able to:

CO1: Understand the concept of mathematical modelling and optimization(Understand)

CO2: Analyze the complexity of solutions to an optimization problem(Analyze)

CO3: Assess the performance of the solutions to an optimization problem (Evaluate)

CO4: Formulate Mathematical models for optimization problems (Evaluate)

CO5: Develop mathematical models to solve the given optimization problem(Create)

Detailed Syllabus:

Module	Topics	No of Lectures
1	Methods of finding initial basic feasible solutions, Tests for optimality. Assignment Problem, Mathematical form of assignment problem, methods of solution.	9

2	Network analysis by linear programming and shortest route, maximal flow problem. Introduction to Non-traditional optimization.	9
3	Computational Complexity – NP-Hard, NP-Complete. Tabu Search- Basic Tabu search, Neighborhood, Candidate list, Short term and Long term memory.	9
4	Genetic Algorithms- Basic concepts, Encoding, Selection, Crossover, Mutation. Simulated Annealing - Acceptance probability, Cooling, Neighborhoods, Cost function.	9
5	Application of GA and Simulated Annealing in solving sequencing and scheduling problems and Travelling salesman problem.	9

Reference Books

1. Optimization: Theory and applications, S. S. Rao, Wiley Eastern Ltd.

Practical:

1. Introduction to Optimization
2. Classical Optimization Techniques
3. Unconstrained Optimization: Elimination Methods
4. Unconstrained Optimization: Interpolation Method
5. Unconstrained Optimization: Direct Root Methods
6. Constrained Optimization: Equality Constraints
7. Constrained Optimization: Inequality Constraints
8. Implement a Genetic algorithm to generate a scheduler

Web Services

*at Semester V
(Implemented during Academic Year 2022-23)*

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Web services basics	9
2	Service Oriented Architecture	9

3	The REST Architectural style	9
4	RESTful web services	9
5	Developing Service-Oriented Applications with WCF	9
	Total	45

Course Objectives:

By the end of the course, learners will be able to:

- Understand and learn Web Services
- Understand XML concepts
- Understand paradigms needed for testing Web Services with REST and WCF
- Analyze the use of WCF Foundations

Course Outcome:

At the end of this course , the students will be able to:

CO1: Create and consume web services to efficiently use market leading environment tools.

(Create)

CO2: Develop web services to identify and select the appropriate framework components.

(Create)

CO3: Discuss and Develop RESTful web services and WCF web services (Create)

CO4: Understand the principles of SOA (Understand)

CO5: Apply and Analyze RESTful web services (Apply)

Detailed Syllabus:

Module	Topics	No of Lectures
1	Web services basics : What Are Web Services? Types of Web Services Distributed computing infrastructure, overview of XML, SOAP, Building Web Services with JAX-WS, Registering and Discovering Web Services	9
2	Service Oriented Architecture, Web Services Development Life Cycle, Web Services Management, Developing and consuming simple Web Services across platform,	9

3	The REST Architectural style : Introducing HTTP, The core architectural elements of a RESTful system, Description and discovery of RESTful web services, Java tools and frameworks for building RESTful web services	9
4	JSON message format and tools and frameworks around JSON, Build RESTful web services with JAX-RS APIs, The Description and Discovery of RESTful Web Services, Design guidelines for building RESTful web services, Secure RESTful web services	9
5	Developing Service-Oriented Applications with WCF : What Is Windows Communication Foundation, Fundamental Windows Communication Foundation Concepts, Windows Communication Foundation Architecture, WCF and .NET Framework Client Profile, Basic WCF Programming, WCF Feature Details.	9

Reference Books

1. Web Services & SOA Principles and Technology, Second Edition, Michael P. Papazoglou.
2. RESTful Java Web Services, JobineshPurushothaman, PACKT Publishing, 2nd Edition, 2015
3. Developing Service-Oriented Applications with WCF, Microsoft, 2017
<https://docs.microsoft.com/en-us/dotnet/framework/wcf/index>
4. Developing Java Web Services, R. Nagappan, R. Skoczylas, R.P. Sriganesh, Wiley India.
5. Developing Enterprise Web Services, S. Chatterjee, J. Webber, Pearson Education.
6. Leonard Richardson and Sam Ruby, RESTful Web Services, O'Reilly, 2007

Practical:

1. Write a program to implement to create a simple web service that converts the temperature from Fahrenheit to Celsius and vice versa.
2. Write a program to implement the operation so it can receive requests and will return a response in two ways. a) One - Way operation b) Request –Response
3. Develop client which consumes web services developed in different platforms.
4. Write a JAX-WS web service to perform the following operations. Define a Servlet / JSP that consumes the web service.
5. Define a web service method that returns the contents of a database in a JSON string. The contents should be displayed in a tabular format.

6. Define a RESTful web service that accepts the details to be stored in a database and performs CRUD operation.
7. Implement a typical service and a typical client using WCF.
8. Use WCF to create a basic ASP.NET Asynchronous JavaScript and XML (AJAX) service.
9. Demonstrates using the binding attribute of an endpoint element in WCF.

Ethical Hacking

at Semester V

(Implemented during Academic Year 2022-23)

Modules at a Glance

Sr. No.	Modules	No. of lectures
1	Introduction to Ethical Hacking, Footprinting and Reconnaissance	9
2	Scanning, Enumeration, Vulnerability Analysis	9

3	System Hacking, Malware Threats, Sniffing	9
4	Denial of Service, Session Hijacking, Evading IDS, Firewall and Honeypots	9
5	SQL Injection and Buffer Overflow, Hacking Mobile Platforms	9
	Total	45

Course Objectives:

By the end of the course, learners will be able to:

- Understand the concept of Ethical Hacking.
- Explain the hacking concepts like Scanning, System hacking, Sniffers, DoS and use tools to implement it.
- Ethically penetrates into network systems using various tools to test the strength of a network.
- Understand the different types of malware with examples.

Course Outcome:

After completing this course learners will be able to:

CO1: Understand the concepts of Ethical Hacking and compare between classes of Hackers.
(Understand, Analyze)

CO2: Describe Scanning and System Hacking along with identifying and using tools to carry out the process. (Apply)

CO3: Demonstrate different tools for understanding the concept of Sniffers, DoS attack, SQL Injection. (Apply)

CO4: Understand and provide justification with examples on malware use and how each works in their way to compromise a system. (Understand, Evaluate)

CO5: Comprehend the dangers associated with penetration testing. (Understand)

Detailed Syllabus:

Module	Topics	No of Lectures
1	Introduction to Ethical Hacking: Information Security Overview; Information Security Threats and Attack Vectors; Penetration Testing Concepts; Hacking Concepts; Ethical Hacking Concepts; Information Security Controls Footprinting and Reconnaissance: What is Footprinting? Terminology in Footprinting, Threats Introduced by Footprinting, Footprinting Process	9
2	Scanning: What is Scanning? Types of Scan, Checking for Live Systems, The Family Tree of Scans, OS Fingerprinting	9

	<p>Enumeration: What Is Enumeration? What Is Meant by Null Sessions? What Is SNMP Enumeration? Windows 2000 DNS Zone Transfer, What Are the Steps Involved in Performing Enumeration?</p> <p>Vulnerability Analysis: Vulnerability Assessment Concepts, Vulnerability Assessment Tools</p>	
3	<p>System Hacking: Understanding Password-Cracking Techniques, Understanding Different Types of Passwords, Understanding Keyloggers and Other Spyware Technologies, Understanding Rootkits, Understanding Steganography Technologies, Understanding How to Cover Your Tracks and Erase Evidence</p> <p>Malware Threats: Malware, Overt and Covert Channels</p> <p>Sniffing: Understanding Sniffers, Using a Sniffer, Switched Network Sniffing</p>	9
4	<p>Denial of Service: Understand the Types of DoS Attacks, Understand How DDoS Attacks Work, Understand How BOTs/BOTNETs Work, What Is a “Smurf” Attack?, What Is “SYN” Flooding?, Describe the DoS/DDoS Countermeasures</p> <p>Session Hijacking: Spoofing vs. Hijacking, Active and Passive Attacks, Session Hijacking and Web Apps, Types of Application-Level Session Hijacking, Network Session Hijacking</p> <p>Evading IDS, Firewall and Honeypots: The Role of Intrusion Detection Systems, Firewalls, What’s That Firewall Running?, Honeypots, Run Silent, Run Deep: Evasion Techniques, Evading Firewalls</p>	9
5	<p>SQL Injection and Buffer Overflow: What Is SQL Injection?, Understand the Steps to Conduct SQL Injection, Understand SQL Server Vulnerabilities, Describe SQL Injection Countermeasures</p> <p>Buffer Overflows, Identify the Different Types of Buffer Overflows and Methods of Detection, Overview of Stack-Based Buffer Overflows, Overview of Buffer Overflow Mutation Techniques</p> <p>Hacking Mobile Platforms: Mobile Platform Attack Vectors, Mobile Platform Vulnerabilities and Risk, Mobile Device Management, Mobile Protection Tools, Mobile Pen Testing</p>	9

Reference Books

1. EC-Council, “Ethical Hacking and Countermeasures Attack Phases”, Cengage Learning
2. Michael Simpson, Kent Backman, James Corley, “Hands-On Ethical Hacking and Network Defense”, Cengage Learning

3. James S. Tiller, “The Ethical Hack: A Framework for Business Value Penetration Testing”, Auerbach Publications, CRC Press
4. Sean-Philip Oriyano, CEH v9 Certified Ethical Hacker Version 9 Study Guide, Wiley Publishing, Inc.
5. Kimberly Graves, CEH Official Certified Ethical Hacker Review Guide, Wiley Publishing, Inc.

Practical:

1. Use Google, traceroute and Whois for Reconnaissance
2. Use Cain and Abel for cracking Windows account password using Dictionary attack and to decode wireless network passwords
3. Implementing cryptanalysis tool.
4. Study and implementation of sniffing tools
5. Study of Denial of Service attack tools.
6. Study of Hijacking tools.
7. Perform SQL injection attack
8. Create a simple keylogger using python
9. Using the tools for scanning network
10. Use Wireshark (Sniffer) to capture network traffic and analyze

Introduction to Big Data Analytics
at Semester V
(Implemented during Academic Year 2022-23)

Modules at a Glance

Sr. No.	Topics	No. of Lectures
1.	Introduction to Big Data	9
2.	Mining data streams	9
3.	Introduction to Big Data Analytics and R Programming	9
4.	Hadoop	9
5.	Frameworks	9
Total		45

Course Objectives:

By the end of the course, learners will be able to:

- Understand the Big Data Platform and its Use cases
- Provide an overview of Apache Hadoop
- Provide HDFS Concepts and Interfacing with HDFS
- Understand Map Reduce Jobs
- Provide hands on Hadoop Eco System
- Apply analytics on Structured, Unstructured Data.
- Exposure to Data Analytics with R.

Course Outcome:

After completing this course learners will be able to:

CO1: Remember the definitions of concepts of Big Data and Business Intelligence Tools.(Remember)

CO2: Understand decision making Theory and Strategies for Big Data. (Understand)

CO3: Understand different Business Intelligence Applications (Understand)

CO4: Knowledge of Decision making using analysis on the Big Data (Remember)

CO5: Identify and study the Big Data Analysis by Decision Theory and Strategy(Understand)

Detailed Syllabus:

Module	Topics	No. of Lectures
1	INTRODUCTION TO BIG DATA: Introduction to Big Data Platform – Challenges of Conventional Systems - Nature of Data Evolution Of Analytic Scalability - Intelligent data analysis- Analytic Processes and Tools - Analysis vs Reporting - Modern Data Analytic Tools - Statistical Concepts: Sampling Distributions - Re-Sampling - Statistical Inference - Prediction Error	9
2	MINING DATA STREAMS: Introduction To Streams Concepts – Stream Data Model and Architecture - Stream Computing - Sampling Data in a Stream – Filtering Streams – Counting Distinct Elements in a Stream – Estimating Moments – Counting Oneness in a Window – Decaying Window - Real time Analytics Platform (RTAP) Applications - Case Studies - Real Time Sentiment Analysis, Stock Market Predictions.	9
3	INTRODUCTION TO BIG DATA ANALYTICS & R PROGRAMMING: Analyzing, Visualization and Exploring the Data, Statistics for Model Building and Evaluation, Introduction to R and RStudio, Basic analysis in R, Intermediate R, Intermediate analysis in R, Advanced Analytics - K-means clustering, Association rules-Speedup, Linear Regression, Logistic Regression, Naïve Bayes, Decision Trees, Time Series Analysis, Text Analysis	9
4	HADOOP: History of Hadoop- The Hadoop Distributed File System – Components of HadoopAnalyzing the Data with Hadoop- Scaling Out- Hadoop Streaming- Design of HDFS-Java interfaces to HDFSBasics- Developing a Map Reduce Application-How Map Reduce Works-Anatomy of a Map Reduce Job run-Failures-Job SchedulingShuffle and Sort – Task execution - Map Reduce Types and Formats- Map Reduce Features	9
5	FRAMEWORKS: Applications on Big Data Using Pig and Hive – Data processing operators in Pig – Hive services – HiveQL – Querying Data in Hive - fundamentals of HBase and ZooKeeper - IBM InfoSphere BigInsights and Streams. Visualizations - Visual data analysis techniques, interaction techniques; Systems and application	9

Reference Books

1. Prajapati, Big Data Analytics with R and Hadoop, 2014.
2. Stephan Kudyba, Big Data, Mining, and Analytics: Components of Strategic Decision Making, Auerbach Publications, March 12, 2014 .
3. Michael Minelli, Michele Chambers, Ambiga Dhiraj , Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses,Wiley Publications,2013.

Additional References

1. Dr. Mark Gardener, Beginning R: The Statistical Programming Language (Wrox), 2013
2. Anand Rajaraman and Jeffrey David Ullman, —Mining of Massive Datasets|, Cambridge University Press, 2012.
3. Bill Franks, —Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics|, John Wiley & sons, 2012.
4. Pete Warden, —Big Data Glossary|, O’Reilly, 2011.
5. Tom White — Hadoop: The Definitive Guide| Third Edition, O’reilly Media, 2012.
6. Zikopoulos, Paul, Chris Eaton, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, Tata McGraw Hill Publications, 2011.

Practical:

1. Install and Setup MongoDB. Also, perform CRUD operations on the given dataset.
2. Create a model to classify the underlying dataset into different groups.
3. Implement Naive Bayes classifier.
4. Create a model to cluster the underlying dataset.
5. Setup and install Hadoop.
6. Performing a MapReduce Job for word search count (look for specific keywords in a file)
7. Install and execute different commands on the Hadoop Distributed File System.
8. Install and execute different commands on Hive.
9. Install and Run Pig then write Pig Latin scripts

Wireless Sensor Network

*at Semester V
(Implemented during Academic Year 2022-23)*

Modules at a Glance

Sr. No.	Modules	No. of Lectures

1	Introduction, Sensor Node Hardware and Network Architecture	9
2	Medium Access Control Protocols for Sensor Networks	9
3	Routing Protocols protocols and strategies for sensor Networks	9
4	Transport layer Protocols design issues	9
5	Sensor Network simulation and Programming approaches	9
	Total	45

Course Objectives:

This course introduces the features of Wireless Sensor Networks and their architecture.

- Understand the basic features of Wireless Sensor networks
- Understand and apply the features of different Wireless sensor Architectures for real world scenarios.
- Understand and apply the protocols of MAC and Network layer for real world Wireless sensor networks
- Sensor Network Programming approaches.
- Design Wireless sensor network for Real time Applications

Course Outcome:

Upon completion of this course, the students will be able to:

CO1: Make critical assessment of wireless networks.(Create)

CO2: Comprehend the fundamentals of wireless networks.(Understand)

CO3: Implement MAC and Network layer protocols for Sensor networks.(Apply)

CO4: Categorize wireless sensor networks.(Understand)

CO5: Simulate Wireless sensor network to understand real life applications.(Create)

Detailed Syllabus:

Module	Topics	No of Lectures
1	Introduction: Ambient intelligence, introducing Wireless sensor Network applications, characteristics, Collaborative processing, Mobile ad hoc networks and wireless sensor networks Key definitions of sensor network. Mobile ad hoc networks and wireless sensor networks.	9

2	Sensor node architecture, The Sensing Subsystem, Analog-to-Digital Converter, Microcontroller, Structure of a Microcontroller, Optimization goals and figures of merit, Design principles for WSNs	9
3	Mac protocol: overview, Medium Access Control Protocol, Contention-Free Medium access, Contention-Based Medium access, Wireless MAC Protocols, Characteristics requirements, Contention-Free and Contention-Based MAC protocols.	9
4	Routing protocols: Network Organization, Route Discovery, Protocol Operation, Routing Metrics, Data-Centric Routing, SPIN family, Routing strategies, Routing methods	9
5	Transport Control Protocols: Traditional Transport Control Protocols, Transport Protocol Design Issues, Examples of Existing Transport Control Protocols, Sensor Network Programming approaches : Challenges in Sensor Network Programming, Node-Centric Programming, Macro programming, Database Approaches, Dynamic Reprogramming, Sensor Network Simulators.	9

Reference Books

1. Holger Karl , Andreas Willig . (2017), Protocols and architectures for Wireless Sensor Networks, Wiley India Pvt. Ltd, New Delhi.
2. Dargie W, Poellabauer C. (2018), Fundamentals of Wireless Sensor Networks: theory and practice, Wiley India Pvt. Ltd, New Delhi
3. Kazem Soharaby, Daniel Minoli, Taieb Znati (2016), Wireless Sensor Networks: Technology, protocols, and Applications, Wiley India Pvt. Ltd, New Delhi

Practical:

1. Create simple client to server wireless simulation.
2. Explore and Understand TinyOs Computational concepts
3. Implement Wireless Sensor Network basics.
4. Create simulation by developing a program to Make LED ON OFF.
5. Create and simulate simple Ad-Hoc Network.
6. Implement Manet Routing
7. Observe Atmospheric pressure from the environment using Atmospheric Pressure Sensor.
8. Create MAC protocol simulation for wireless sensor network.
9. Create Routing Protocol simulation for Wireless sensor Network.
10. Build Smart home/ smart city using sensor technology.

Game Programming

*at Semester V
(Implemented during Academic Year 2022-23)*

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Mathematics for Computer Graphics	9
2	Transformations	9
3	DirectX Pipeline and Programming	9
4	Interpolation and Character Animation	9
5	Unity Engine and XR	9
	Total	45

Course Objectives:

By the end of the course, learners will be able to:

- Introduce the use of the components of a graphics system and become familiar with building the approach of graphics system components and algorithms related with them.
- Discuss the application of computer graphics concepts in the development of computer games, information visualization, and business applications.
- Develop creativity and individuality in problem solving and performing tasks
- Improve their skills and knowledge related to specific job positions individually
- Introduce them to scientific work in the areas of virtual reality, computer graphics, artificial intelligence

Course Outcome:

At the end of the course the learner should be able to

CO1: Understand basic concepts of Computer Graphics. (Level: Understanding)

CO2: Solve various Transformation on various Objects. (Level: Apply)

CO3: Explain the functioning of DirectX. (Level: Understanding)

CO4: Explain the fundamentals of animation, virtual reality and its related technologies (Level: Understanding)

CO5: Develop 2D and 3D Games using Unity. (Level: Create)

Detailed Syllabus:

Module	Topics	No of Lectures
1	Cartesian Coordinate system: The Cartesian XY-plane, Function Graphs, Geometric Shapes, Polygonal Shapes, Areas of Shapes, Theorem of	9

	<p>Pythagoras in 2D, Coordinates, Theorem of Pythagoras in 3D, 3D Polygons, Euler's Rule</p> <p>Vectors: Vector Manipulation, multiplying a Vector by a Scalar, Vector Addition and Subtraction, Position Vectors, Unit Vectors, Cartesian Vectors, Vector Multiplication, Scalar Product, Example of the Dot Product, The Dot Product in Lighting Calculations, The Dot Product in Back-Face Detection, The Vector Product, The Right-Hand Rule, deriving a Unit Normal Vector for a Triangle Areas, Calculating 2D Areas</p>	
2	<p>Transformations: 2D Transformations, Matrices, Homogeneous Coordinates, 3D Transformations, Change of Axes, Direction Cosines, rotating a Point about an Arbitrary Axis, Transforming Vectors, Determinants, Perspective Projection, Interpolation</p>	9
3	<p>DirectX: Understanding GPU and GPU architectures. How are they different from CPU Architectures? Understanding how to solve by GPU?</p> <p>Introduction to DirectX 11: COM, Textures and Resources Formats, the swap chain and Page flipping, Depth Buffering, Texture Resource Views, Multisampling Theory and MS in Direct3D, Feature Levels</p> <p>Direct3D 11 Rendering Pipeline: Overview, Input Assembler Stage (IA), Vertex Shader Stage (VS), The Tessellation Stage (TS), Geometry Shader Stage (GS), Pixel Shader Stage (PS), Output merger Stage (OM) Understanding Meshes or Objects, Texturing, Lighting, Blending.</p>	9
4	<p>Trigonometry: The Trigonometric Ratios, Inverse Trigonometric Ratios, Trigonometric Relationships, The Sine Rule, The Cosine Rule, Compound Angles, Perimeter Relationships</p> <p>Interpolation: Linear Interpolant, Non-Linear Interpolation, Trigonometric Interpolation, Cubic Interpolation, Interpolating Vectors, Interpolating Quaternions</p> <p>Curves: Circle, Bezier, B-Splines</p> <p>Analytic Geometry: Review of Geometry, 2D Analytic Geometry, Intersection Points, Point in Triangle, and Intersection of circle with straight line</p>	9
5	<p>Unity Engine: Multi-platform publishing, VR + AR: Introduction and working in Unity, 2D, Graphics, Physics, Scripting, Animation, Timeline, Multiplayer and Networking, UI, Navigation and Pathfinding, XR, Publishing.</p> <p>Scripting: Scripting Overview, Scripting Tools and Event Overview</p> <p>XR: VR, AR, MR, Conceptual Differences. SDK, Devices</p>	9

Reference Books

1. Mathematics for Computer Graphics, John Vince, Springer-Verlag London, 5th Edition,2017
2. Mathematics for 3D Game Programming and Computer Graphic, Eric Lengyel, Delmar Cengage Learning, Delmar Cengage Learning,2011
3. Introduction To 3D Game Programming with DirectX® 11, Frank D Luna, Mercury Learning and Information,2012.
4. Computer Graphics, C Version, Donald Hern and Pauline Baker, Pearson Education, 2nd Edition, 1997
5. HLSL Development Cookbook, Doron Feinstein, PACKT Publishing,2013
6. <https://docs.unity3d.com/Manual/index.html> - Free

Practical:

Setup DirectX 11, Window Framework and Initialize Direct3D Device

1. Draw a triangle using Direct3D 11 (Buffers, Shaders and HLSL)
2. Texture the Triangle using Direct 3D 11 (Texturing)
3. Programmable Diffuse Lightning using Direct3D 11 (Lightning)
4. Programmable Spot Lightning using Direct3D 11(Specular Lightning)
5. Loading models into DirectX 11 and rendering.

Perform following Practical using online content from the Unity Tutorials Web--sites:
<https://unity3d.com/learn/tutorials/s/interactive-tutorials>

1. 2D UFO
1. Space Shooter
1. Roll a Ball

Web Technologies

*at Semester V
(Implemented during Academic Year 2022-23)*

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	Introducing .NET	9
2	Web Form Fundamentals	9
3	Error Handling, Logging, and Tracing, State Management, Styles, Themes, and Master Pages	9
4	ADO.NET Fundamentals	9
5	XML, Security Fundamentals, ASP.NET Ajax	9
	Total	45

Course Objective:

By the end of the course, learners will be able to:

1. Understand basic building blocks of Dot Net.
2. Assimilate C# Fundamentals, Exception handling, Design Interfaces and Collections in C#.
3. Defines and discuss major concepts, tool, techniques, and methods of web application development.
4. Create web application using ASP.NET.
5. Implement the the database connectivity with ASP.NET.

Course outcome:

After completing this course learners will be able to:

CO1: Understand code solutions and compile C# projects within the .NET framework.

(Understand)

CO2: Develop simple file test assembly.(Create)

CO3: Apply and Create GUI components in C#. Design and Implement Web Applications, Control Library, Advanced UI Programming & Data Binding concepts.(Create, Apply)

CO4: Design and Implement database connectivity using ADO.NET, XML in C#.NET specifically ADO.NET.(Create)

CO5: Develop partial refreshes of web pages using ajax(Create)

Detailed Syllabus:

Module	Topics	No of Lectures
1	<p>Introducing .NET: The .NET Framework, C#, VB, and the .NET Languages, The Common Language Runtime, The .NET Class Library. The C# Language, C# Language Basics, Variables and Data Types, Variable Operations, Object-Based Manipulation, Conditional Logic, Loops, Methods. Types, Objects, and Namespaces: The Basics About Classes, Building a Basic Class, Value Types and Reference Types, Understanding Namespaces and Assemblies</p>	9
2	<p>Web Form Fundamentals: Writing Code, Using the Code-Behind Class, Adding Event Handlers, Understanding the Anatomy of an ASP.NET Application, Introducing Server Controls, Using the Page Class, Using Application Events, Configuring an ASP.NET Application. Form Controls, Stepping Up to Web Controls, Web Control Classes, List Controls, Table Controls, Web Control Events and AutoPostBack, Validation, Understanding Validation, Using the Validation Controls, Rich Controls, The Calendar, The AdRotator, Pages with Multiple Views, User Controls and Graphics, User Controls, Dynamic Graphics, The Chart Control, Website Navigation, Site Maps, URL Mapping and Routing, The SiteMapPath Control, The TreeView Control, The Menu Control.</p>	9
3	<p>Error Handling, Logging, and Tracing : Avoiding Common Errors, Understanding Exception Handling, Handling Exceptions, Throwing Your Own Exceptions, Using Page Tracing</p> <p>State Management, Understanding the Problem of State, Using View State, Transferring Information Between Pages, Using Cookies, Managing Session State, Configuring Session State, Using Application State, Comparing State Management Options</p> <p>Styles, Themes, and Master Pages: Styles, Themes, Master Page Basics, Advanced Master Pages.</p>	9
4	<p>ADO.NET Fundamentals: Understanding Databases, Configuring Your Database, Understanding SQL Basics, Understanding the Data Provider Model, Using Direct Data Access, Using Disconnected Data Access. Data Binding: Introducing DataBinding, Using Single-Value Data Binding, Using Repeated-Value Data Binding, Working with Data Source Controls. The Data Controls: The GridView, Formatting the GridView, Selecting a GridView Row, Editing with the GridView, Sorting and Paging the GridView, Using GridView Templates, The DetailsView and FormView</p>	9
5	<p>XML: The XML Classes, XML Validation, XML Display and Transforms.</p> <p>Security Fundamentals: Understanding Security Requirements, Authentication and Authorization, Forms Authentication, Windows Authentication.</p> <p>ASP.NET AJAX: Understanding Ajax, Using Partial Refreshes, Using Progress Notification, Implementing Timed Refreshes, Working with the ASP.NET AJAX Control Toolkit.</p>	9

Reference Books

1. Beginning ASP.NET 4.5 in C#, Matthew MacDonald , Apress , 2012
2. C# 2015, Anne Bohem and Joel Murach , Murach , Third , 2016
3. Murach's ASP.NET 4.6 Web Programming in C#2015 , Mary Delamater and Anne Bohem , SPD , Sixth , 2016
4. ASP.NET 4.0 Programming, J.Kanjilal , Tata McGraw-Hill , 2011
5. Programming ASP.NET, D.Esposito , Micosoft Press (Dreamtech) , 2011
6. Beginning Visual C# 2010 , K. Watson , C. Nagel, J.H Padderson, J.D. Ried, M Skinner, Wrox (Wiley) ,2010

Practical:

1. Working with basic C# and ASP.NET

- a) Create an application that obtains four int values from the user and displays the product.
- b) Create an application to demonstrate string operations.
- c) Create an application that receives the (Student Id, Student Name, Course Name, Date of Birth) information from a set of students. The application should also display the information of all the students once the data entered.
- d) Create an application to demonstrate following operations
 - i. Generate Fibonacci series.
 - ii. Test for prime numbers.
 - iii. Test for vowels.
 - iv. Use of foreach loop with arrays
 - v. Reverse a number and find sum of digits of a number.

2. Working with Object Oriented C# and ASP .NET

- a) Create a simple application to perform following operations
 - i. Finding factorial Value
 - ii. Money Conversion
 - iii. Quadratic Equation
 - iv. Temperature Conversion
- b) Create a simple application to demonstrate use of following concepts
 - i. Function Overloading
 - ii. Inheritance (all types)
 - iii. Constructor overloading
 - iv. Interfaces
- c) Create a simple application to demonstrate use of the following concepts
 - i. Using Delegates and events
 - ii. Exception handling

3. Working with Web Forms and Controls

- a) Create a simple web page with various server controls to demonstrate setting and use of their properties.(Example,AutoPostBack)
- b) Demonstrate the use of Calendar control to perform following operations.
 - i) Display messages in a calendar control
 - ii) Display vacation in a calendar control
 - iii) Selected day in a calendar control using style

iv) Difference between two calendar date

c) Demonstrate the use of Treeview control and perform following operations. i) Treeview Control and datalist ii) Exception handling

4. Working with Form Controls

a) Create a Registration form to demonstrate use of various Validation controls.

b) Create Web Form to Demonstrate use of Adrotator Control.

c) Create Web Form to demonstrate use User Controls

5. Working with Navigation , Beautification and Master page.

a) Create Web Form to demonstrate use of Website Navigation controls and Site map.

b) Create a web application to demonstrate use of Master Page with applying Styles and Themes for page beautification

c) Create a web application to demonstrate various states of ASP.NET Pages.

6. Working with Database

a) Create a web application bind data in a multiline textbox by querying in another textbox

b) Create a web application to display records by using a database.

c) Demonstrate the use of Datalist link control.

7. Working with Database

a) Create a web application to display Data Binding using dropdownlist control.

b) Create a web application to display the phone no of an author using a database.

c) Create a web application for inserting and deleting records from a database. (Using Execute-Non Query).

8. Working with data controls

a) Create a web application to demonstrate various uses and properties of SqlDataSource

b) Create a web application to demonstrate data binding using DetailsView and FormView Controls.

c) Create a web application to display Using Disconnected Data Access and Data Binding using GridView.

9. Working with GridView control a) Create a web application to demonstrate use of GridView control template and GridView hyperlink.

b) Create a web application to demonstrate use of GridView button column and GridView Events.

c) Create a web application to demonstrate GridView paging and Creating own table format using GridView

10. Working with AJAX and XML

a) Create a web application to demonstrate reading and writing operations with XML.

b) Create a web application to demonstrate Form Security and Windows Security with proper Authentication and Authorization Properties.

c) Create a web application to demonstrate use of various Ajax controls.

Evaluation Scheme

Internal Exam-25 Marks

- **Test– 20 Marks which will be converted out of 10 Marks-** Duration 40 mins

It will be conducted either as a written test or using any open source learning management system such as Moodle (Modular Object-Oriented Dynamic Learning Environment) or a test based on an equivalent online course on the contents of the concerned course (subject) offered by or build using MOOC (Massive Open Online Course) platform.

- **Assignments/Presentation/Projects – 10 Marks**

Subject specific Term Work Module/assessment modes –as decided by the department in the beginning of the semester (like Extension/field/experimental work, Short Quiz; Objective test, lab practical, open book test etc. and written assignments, Case study, Projects, Posters and exhibits etc. for which the assessment is to be based on class presentations wherever applicable)

- **Active participation in routine class instructional deliveries - 5 Marks**

Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

External Examination- 75 Marks

Duration - 2.5 Hours.

Theory question paper pattern,-

All questions are compulsory.		
Question	Based on	Marks
Q.1	Unit 1	15
Q.2	Unit 2	15
Q.3	Unit 3	15
Q.4	Unit 4	15
Q.5	Unit 5	15

All questions shall be compulsory with internal choice within the questions.

Each Question may be sub-divided into sub questions as a, b, c, d & e, etc& the allocation of Marks depends on the weightage of the topic.

Practical Examination – 50 marks (Duration: 2 Hours)

Each practical course carries 50 Marks , 40 marks + 05 marks (journal)+ 05 marks(viva)

Minimum 75% practical from each core/allied course are required to be completed and written in the journal.

(Certified Journal is compulsory for appearing at the time of Practical Exam)

***Syllabus of Courses of
Bachelor of Science in Computer Science Programme
at Semester VI
(Implemented during Academic Year 2022-23)***

Building Blocks of Artificial Intelligence

*at Semester VI
(Implemented during Academic Year 2022-23)*

Modules at a Glance

Sr. No.	Modules	No. of lectures
1	Problem solving methods	9
2	Concept learning and decision trees	9
3	Bayesian and Computational learning	9
4	Analytical learning and reinforcement learning	9
5	Mining social network graphs	9
	Total	45

Course Objectives:

By the end of the course, learners will be able to:

- Introduce students to the basic concepts and techniques of Machine Learning.
- Have a thorough understanding of the Supervised and Unsupervised learning techniques
- Study the various probability based learning techniques
- Understand graphical models of machine learning algorithms
- Study about Basic Social Networks concepts, Clustering in social network social circles and Applications

Course Outcome:

After completing this course learners will be able to:

- CO1: Create the learning techniques with this basic knowledge. (Create)
- CO2: Understand Bayesian techniques and derive effectively learning rules.(Understand)
- CO3: Understand and differentiate reinforcement and analytical learning techniques(Understand)
- CO4: Visualize the data and follow of ethic(Understand)
- CO5: Analyse about Ethical Dilemmas & Social Networks(Analyse)
- CO6: Analyse about Social Networks effectiveness (Analyse)

Detailed Syllabus:

Module	Topics	No of Lectures
1	PROBLEM SOLVING METHODS: Problem solving Methods - Search Strategies- Uninformed - Informed - Heuristics - Local Search Algorithms and Optimization Problems - Searching with Partial Observations - Constraint Satisfaction Problems – Constraint Propagation - Backtracking Search - Game Playing -Optimal Decisions in Games -Alpha--Beta Pruning -Stochastic Games	9
2	Machine Learning Techniques: CONCEPT LEARNING AND DECISION TREES: Learning Problems – Designing Learning systems, Perspectives and Issues – Concept Learning – Version Spaces and Candidate Elimination Algorithm – Inductive bias – Decision Tree learning – Representation – Algorithm – Heuristic Space Search	9
3	BAYESIAN AND COMPUTATIONAL LEARNING: Bayes Theorem – Concept Learning – Maximum Likelihood – Minimum Description Length Principle – Bayes Optimal Classifier – Gibbs Algorithm – Naïve Bayes Classifier– Bayesian Belief Network – EM Algorithm – Probably Learning – Sample Complexity for Finite and Infinite Hypothesis Spaces – Mistake Bound Model	9
4	ANALYTICAL LEARNING AND REINFORCED LEARNING: Perfect Domain Theories – Explanation Based Learning – Inductive-Analytical Approaches - FOCL Algorithm – Reinforcement Learning – Task – Q-Learning – Temporal Difference Learning	9
5	MINING SOCIAL NETWORK GRAPHS: Social networks as graphs, Clustering of graphs, Direct discovery of communities in graphs, Partitioning of graphs, Neighbourhood properties in graphs, Data Visualization: Basic principles, ideas and tools for data visualization.	9

Reference Books

1. Machine Learning Tom M. Mitchell McGraw-Hill Education 2013
2. Introduction to Machine Learning Ethem Alpaydin PHI Learning Pvt. 2nd Ed., 2013
3. Machine Learning: A Probabilistic Perspective Kevin P. Murphy 2013
4. W. de NOOY, A. MRVAR, V. BATAGELJ (2018), Exploratory Social Network Analysis with Pajek, Cambridge University Press, 3rd Edition.

Practical:

1. Implement Search Strategies: Breadth first and Depth First search algorithm , Iterative deepening search , A* search algorithm
[One may use data sets like Caravan, Smarket, Weekly, Auto and Boston]
2. For a given data set, do the following:
 - (i) Fit a classification tree
 - (ii) Fit a regression tree
3. Fit a classification model using K Nearest Neighbour (KNN) Algorithm on a given dataset.
4. Generate a regression model and interpret the result for a given data set.
5. Generate forecasting model and interpret the result for a given data set.
6. Create a Bayesian network for a given narrative. Set findings and ask queries [One may use narratives like ‘chest clinic narrative’ and package gRain for the purpose].
7. Implement EM algorithm.
8. Compute the following node level measures:
 - (i) Density; (ii) Degree; (iii) Reciprocity; (iv) Transitivity; (v) Centralization; (vi) Clustering.
9. For a given network find the following:
 - (i) Length of the shortest path from a given node to another node; (ii) the density of the graph; (iii) Draw egocentric network of node G with chosen configuration parameters.
10. Create sociograms for the persons-by-persons network and the committee-by-committee network for a given relevant problem. Create a one-mode network and two-node network for the same.

Data Science
at Semester VI
(Implemented during Academic Year 2022-23)

Modules at a Glance

Sr. No.	Modules	No. of lectures
1	Introduction to Data Science and Data Preprocessing	9
2	Exploratory Data Analysis and Statistical Modeling	9
3	Supervised and Unsupervised Learning	9
4	Semi-structured systems	9
5	Unstructured and Big Data Analytics	9
	Total	45

Course Objectives:

By the end of the course, learners will be able to:

- Understand data science and its processes.
- Understand the data science algorithms and models
- Develop research interest towards advances in data science.
- Explore, sort and analyze megadata from various sources to take advantage of them and reach conclusions to optimize business processes or for decision support.

Course Outcome:

After completing this course learners will be able to:

- CO1: Understand the basic concepts of Data Science, Data Cleansing, Data Curation(Understand)
- CO2: Analyze and understand the situations where a given data science algorithm can be applied(Analyse)
- CO3: Evaluate the performance of different data science models in order to achieve the given objective(Evaluate)
- CO4: Use the standard data science models/process in order to solve the given problem.(Apply)
- CO5: Create innovative models/methodologies to solve the given problem(Create)

Detailed Syllabus:

Module	Topics	No of Lectures
1	Introduction to Data Science: What is Data? Different kinds of data, Data Science Process or lifecycle. Data Preprocessing: Descriptive Data Summarization, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization and Concept Hierarchy Generation	9
2	Exploratory Data Analysis (EDA): Measures of central tendency and dispersion, Bar plot, histogram, Box plots, stem-leaf diagram, multidimensional modeling. Statistical Modeling and Machine Learning Algorithms: Introduction to model selection: Regularization, bias/variance tradeoff e.g.parsimony, AIC, BIC, Cross validation	9
3	Supervised Learning: Regression, linear models, Regression trees, Time-series Analysis, Forecasting, Classification: classification trees, Logistic regression, separating hyperplanes, k-NN Unsupervised Learning: Principal Components Analysis (PCA), k-means clustering, Hierarchical clustering, Density-Based Methods, Grid-Based Methods, Model-Based Clustering Methods, Clustering High-Dimensional Data, Constraint-Based Cluster Analysis, Outlier Analysis, Association rules from frequent itemsets. Ensemble methods: Increasing the Accuracy, Model Selection.	9
4	Semi-structured systems: Semi-structured data Model, management and querying of data.	9
5	Unstructured data analytics systems: Unstructured data model, NoSQL databases, Text Analytics Big data Analytics: What is Big data? ,Document shingling	9

Reference Books

1. Doing Data Science, Rachel Schutt and Cathy O’Neil, O’Reilly.
2. J. Han and M. Kamber, ” Data Mining: Concepts and Techniques”, Second Edition, Elsevier.

Practical:

1. Practical of Data collection, Data curation and management for Unstructured data (NoSQL)
2. Practical of Data collection, Data curation and management for Large-scale Data system (such as MongoDB)
3. Practical of Principal Component Analysis
4. Practical of Clustering
5. Practical of Time-series forecasting
6. Practical of Simple/Multiple Linear Regression
7. Practical of Logistics Regression
8. Practical of Hypothesis testing
9. Practical of Analysis of Variance
10. Practical of Decision Tree

Cloud Computing
at Semester VI
(Implemented during Academic Year 2022-23)

Modules at a Glance

Sr. No.	Modules	No. of lectures
1	Introduction	9
2	Virtualization	9
3	Cloud Computing Services	9
4	Cloud Implementation, Programming and Mobile Cloud Computing	9
5	Exploring the Components of Amazon Web Services	9
	Total	45

Course Objectives:

By the end of the course, learners will be able to:

- Basics of cloud computing.
- Key concepts of virtualization.
- Different Cloud Computing services
- Cloud Implementation, Programming and Mobile cloud computing
- Key components of Amazon Web Services

Course Outcome:

After completion of the course, learners will be able to:

- CO1: Explain the concept Cloud Computing and the different Cloud service and deployment models. (Level: Understand)
- CO2: Describe the importance of virtualization along with their technologies. (Level: Understand)
- CO3: Implement Virtualization using different types of Hypervisors (Level: Apply)
- CO4: Examine and use different cloud computing services. (Level: Analyse)
- CO5: Analyse the components of open stack & Google Cloud platform and understand Mobile Cloud Computing. (Level: Analyse)
- CO6: Analyse and understand the functioning of different components involved in Amazon web services cloud platform. (Level: Analyse)

Detailed Syllabus:

Module	Topics	No of Lectures
1	Defining Cloud Computing, Cloud and other similar configurations, Components of Cloud Computing, Cloud types: NIST and Cloud Cube Models, Cloud Deployment Models and Service Models, Cloud computing architecture, Advantages and Disadvantages of Cloud Computing.	9
2	Characteristics of virtualized environment, Understanding the importance of Hypervisors, Type I & Type II Hypervisors, Taxonomy of virtualization, Implementation Levels of Virtualization, Virtualization of CPU, Memory and I/O Devices, Virtualization and Cloud Computing, Pros and Cons of virtualization, Technology Examples: KVM, Xen, VMware and Hyper-V	9
3	Exploring Cloud Computing Services: SPI Model: Software as a service, Platform as a service, and Infrastructure as a service. Anything as a service or Everything as a service (XaaS): Security as a Service, Identity management as a Service, Database as a Service, Storage as a Service, Collaboration as a Service, Compliance as a Service, Monitoring as a Service, Communication as a Service, Network as a Service, Disaster recovery as a service, Analytics as a Service, Backup as a Service	9
4	OpenStack Cloud Architecture: Feature of Open stack, Components of Open stack, mode of operations. Programming support for Google apps engine- GFS, Big Tables, Chubby, Google APIs. Mobile Cloud Computing: Definition, architecture, benefits and challenges of mobile cloud computing	9
5	AWS cloud computing Platform, Elastic Compute Cloud (EC2): Compute Basics, Instance types, Life cycle of instances. Simple Storage Service (S3): Basics and Operations, Features, Amazon Glacier, Glacier vs S3. Elastic Block Storage (EBS): Basics and Types of EBS Volumes d) Amazon Virtual Private Cloud (Amazon VPC): Subnets, Route tables, Elastic IP Addresses (EIP), Elastic Network Interfaces (ENIs) & Security groups & ACL. Exploring Elastic Load Balancing (ELB): Basics, Types of load balancers, Configuring Elastic Load Balancing, Basics of Cloud Watch & Auto Scaling.	9

Reference Books

1. Barrie Sosinsky, “Cloud Computing Bible”, Wiley Publication.
2. Kailash Jayaswal, Jagannath Kallalurchi, Donald J. Houde, Dr. Deven Shah, “Cloud Computing Black Book”, Dreamtech Press.
3. Joe Baron et.al, “AWS certified solution Architect”, Sybex publication.
4. Mastering Cloud Computing, Rajkumar Buyya, MGH publication

Practical:

1. Creating and running virtual machines on Hosted Hypervisors Type 1 - KVM.
2. Creating and running virtual machines on Hypervisors Type 0 – Xen.
3. Creating and running virtual machines on Hypervisors Type 0 – VMware ESXI
4. Creating and running virtual machines on Hypervisors Type 0 – Hyper-V Installation.
5. To Demonstrate Platform as a Service using Google App Engine.
6. Explore Storage as a service using ownCloud.
7. To demonstrate installation and Configuration of OpenStack Private cloud. Like auto scaling, elastic load balancing, virtual private computing & Networking. Security service provided by Amazon web services. Accessing AWS using web services API provided by Amazon.

Any other practical covering syllabus can be implemented.

Cyber Forensics and Investigations
at Semester VI
(Implemented during Academic Year 2022-23)

Modules at a Glance

Sr. No.	Modules	No. of lectures
1	Introduction to Computer Forensics and data acquisitions	9
2	Network Forensics and Mobile device Forensics	9
3	Email Investigation and Web browser forensics	9
4	Computer Forensics tools and Introduction to Digital Forensics	9
5	Conducting Digital Investigations	9
	Total	45

Course Objectives:

By the end of the course, learners will be able to:

- The course is designed to understand the fundamentals and different methodologies used in Cyber Forensics.
- Understand different types of forensics.
- Study the concept of Digital Evidence and the process of its investigation.
- Emphasize the fundamentals and importance of digital forensics.

Course Outcome:

At the end of the course, learners will be able to:-

- CO1: Define and use different Computer Forensics tools used for investigating. (Remember,Apply)
 CO2: Understand and analyze different Computer Forensics systems for problems. (Understand)
 CO3: Understand Network Forensics and perform data acquisitions. (Understand)
 CO4: Discover various tools to perform forensics. (Understand)
 CO5: Compare with different case studies on the real time scenarios (Analyze)

Detailed Syllabus:

Module	Topics	No of Lectures
1	Computer Forensics: Computer Forensics and Investigation Processes, Understanding Computing Investigations, The Investigator's Office and Laboratory, Data Acquisitions	9
2	Virtual Machines, Network Forensics, and Live Acquisitions:	9

	<p>Virtual Machines Overview. Network Forensics Overview, securing a Network, Performing Live Acquisitions, Developing Standard Procedures for Network Forensics, Reviewing Network Logs, Using Network Tools, Using Packet Sniffers.</p> <p>Cell phone and mobile device forensics: Understanding mobile device forensics, Mobile phone basics, inside mobile devices, inside PDAs, Understanding acquisition procedures for cell phones and mobile devices, Mobile forensics equipment.</p>	
3	<p>E-mail Investigations: Exploring the Role of E-mail in Investigations, Exploring the Roles of the Client and Server in E-mail, Investigating E-mail Crimes and Violations, Understanding E-mail Servers, Using Specialized E-mail Forensics Tools.</p> <p>Web Browser Forensics: Introduction, Cookie Storage and Analysis, Analyzing Cache and temporary internet files, Web browsing activity reconstruction</p>	9
4	<p>Current Computer Forensics Tools:</p> <p>Evaluating Computer Forensics Tool Needs, Types of Computer Forensics Tools, Tasks Performed by Computer Forensics Tools, Computer Forensics Software Tools, Command-Line Forensics Tools, UNIX/Linux Forensics Tools, Other GUI Forensics Tools, Computer Forensics Hardware Tools, Forensic Workstations, Using a Write-Blocker.</p> <p>Digital Forensics: Foundations of Digital Forensics, Digital Evidence, Digital Forensics: Past, Present, and Future, Principles of Digital Forensics, Challenging Aspects of Digital Evidence, Following the Cyber trail. Language of Computer Crime Investigation, The Role of Computers in Crime.</p>	9
5	<p>Conducting Digital Investigations: Digital Investigation Process Models, Scaffolding for Digital Investigations, Applying the Scientific Method in Digital Investigations, Investigative Scenario: Security Breach. Handling a Digital Crime Scene- Published Guidelines for Handling Digital Crime Scenes, Fundamental Principles, Authorization, Preparing to Handle Digital Crime Scenes, Surveying the Digital Crime Scene, Preserving the Digital Crime Scene.</p>	9

Reference Books

1. Bill Nelson, Amelia Phillips, Christopher Steuart, "Guide to Computer Forensics and Investigations", Fourth Edition, Course Technology.
2. Digital Evidence and Computer Crime, Eoghan Casey, Third Edition, 2011, Elsevier Inc.

Practical:

1. Creating a Forensic Image using FTK Imager/Encase Imager : Creating Forensic Image, Check Integrity of Data, Analyze Forensic Image
2. Data Acquisition:
 - Perform data acquisition using:
 - USB Write Blocker + Encase Imager
 - SATA Write Blocker + Encase Imager
 - Falcon Imaging Device
3. Forensics Case Study: Solve the Case study (image file) provide in lab using Encase Investigator or Autopsy
4. Capturing and analyzing network packets using Wireshark (Fundamentals) :
 - Identification the live network
 - Capture Packets
 - Analyze the captured packets
5. Recovering and Inspecting deleted files
 - Check for Deleted Files
 - Recover the Deleted Files
 - Analyzing and Inspecting the recovered filesPerform this using recovery option in ENCASE and also Perform manually through command line
6. Acquisition of Cell phones and Mobile devices
7. Email Forensics
 - Mail Service Providers
 - Email protocols
 - Recovering emails
 - Analyzing email header
8. Web Browser Forensics
 - Web Browser working
 - Forensics activities on browser
 - Cache / Cookies analysis
 - Last Internet activity
9. File System Analysis using The Sleuth Kit
10. Using Steganography tools for computer forensics.

Mobile and Pervasive Computing
at Semester VI
(Implemented during Academic Year 2022-23)

Modules at a Glance

Sr. No.	Modules	No. of Lectures
1	History – Wireless communications:	9
2	Overview of a Modern 4G Telecommunications System	9
3	Advances in 4G Communication Networks: A 5G PERSPECTIVE	9
4	Introduction to Ubiquitous Computing	9
5	Pervasive Computing Devices Smart Environment:	9
	Total	45

Course Objectives:

By the end of the course, learners will be able to:

- Explain the basics of mobile telecommunication system.
- Enhance the knowledge about 4G and 5G.
- List and exemplify the key technologies involved in the development Ubicomp Systems
- Develop an attitude to identify and propose solutions for security and privacy issues
- Explore the trends and problems of current pervasive computing systems using examples.

Course Outcome:

After completing this course learners will be able to:

CO1:Develop the fundamental theoretical concepts in pervasive computing.(Create)

CO2:Assess the aspects of context awareness(Evaluate)

CO3:Apply the methods for efficient resource allocation and task migration(Apply)

CO4:Analyze the HCI Service Selection and HCI migration framework(Analyze)

CO5:Apply and implement pervasive application systems(Apply)

Detailed Syllabus:

Module	Topics	No of Lectures
1	History – Wireless communications: GSM – DECT – TETRA – UMTS – IMT – 2000 – Bluetooth, WiFi, WiMAX, 3G ,WATM.- Mobile IP protocols -WAP push architecture-Wml scripts and applications. Data networks – SMS – GPRS – EDGE – Hybrid Wireless100 Networks – ATM – Wireless ATM.	9
2	OVERVIEW OF A MODERN 4G TELECOMMUNICATIONS SYSTEM Introduction. LTE-A System Architecture. LTE RAN. OFDM Air Interface. Evolved PacketCore. LTE Requirements. LTE-Advanced. LTE-A in Release. OFDMA – Introduction. OFDM Principles. LTE Uplink—SC-FDMA. Summary of OFDMA.	9
3	ADVANCES IN 4G COMMUNICATION NETWORKS: A 5G PERSPECTIVE Introduction - Evolution toward 5G Networks - Challenges in 5G Networks – Emerging Trends in 5G Networks - LTE/LTE-A 4G and Beyond Technology - MIMO Enhancements:3D -Beamforming, Full-Dimension - MIMO, and Massive MIMO - Millimeter-Wave Communication Technology - Channel State Information Feedback Concepts of 3GPP LTE - Channel State Information Feedback Concepts for 5G.	9
4	Introduction to Ubiquitous Computing Concept of Distributed Computing, Mobile Computing, Pervasive Computing, Wearable Computing, Modeling the Key Ubiquitous/Pervasive Computing Properties, Mobile Adaptive Computing , Mobility Management and Caching.	9
5	Pervasive Computing Devices Smart Environment: CPI and CCI Smart Devices: Application and Requirements, Device Technology and Connectivity, Human Computer Interaction.- Explicit HCI, Implicit HCI, User Interface and Interaction for four hand-held widely used devices, Hidden UI via basic smart devices, Hidden UI via wearable and Implanted devices, Human centered design, user models.	9

Reference Books

1. Jochen H. Schiller, —Mobile Communications, Second Edition, Pearson Education, New Delhi, 2007.
2. Juha Korhonen, —Introduction to 4G Mobile Communications, Artech House Publishers, 2014.
3. M. Bala Krishna, Jaime Lloret Mauri, —Advances in Mobile Computing and Communications: Perspectives and Emerging Trends in 5G Networks, CRC 2016
4. Prasant Kumar Pattnaik, Rajib Mall, —Fundamentals of Mobile Computing, PHI Learning Pvt. Ltd, New Delhi – 2012.
5. Minyi Guo, Jingyu Zhou, Feilong Tang, Yao Shen, ”Pervasive Computing: Concepts, Technologies and Applications”, CRC Press, 2016.
6. Obaidat, Mohammad S., Mieso Denko, and Isaac Woungang, eds. Pervasive computing and networking. John Wiley & Sons, 2011.
7. Laurence T. Yang, Handbook On Mobile And Ubiquitous Computing Status And Perspective, 2012, CRC Press.

Practicals:

1. STUDY EXPERIMENT To explore overall view about Pervasive Computing Architecture , Communication protocols , Software infrastructure and Security mechanisms
2. To perform Localization concepts in Android
3. To perform Road and Traffic sensing in Android
4. Applications for location-based messages in Android
5. To perform Global Positioning system in Android
6. Case study of Class Room 2020
7. Case study of Super Market
8. Case study of Hospital Management
9. Case Study: iCampus Prototype,
10. Case Study: IPSpace: An IPv6-Enabled Intelligent Space

Digital Image Processing
at Semester VI
(Implemented during Academic Year 2022-23)

Modules at a Glance

Sr. No.	Modules	No. of lectures
1	Introduction, Convolution and Correlation	9
2	Image Transformations and Enhancement	9
3	Binary and Color Image Processing	9
4	Image Compression and Watermarking	9
5	Image Segmentation	9
	Total	45

Course Objectives:

By the end of the course, learners will be able to:

- Study the image fundamentals and mathematical transforms necessary for image processing.
- Study the image enhancement techniques
- Study image restoration procedures.
- Study the image compression procedures.

Course Outcome:

After completion of the course, learners will be able to:

CO1: Review the fundamental concepts of a digital image processing system. (Level: Understand)

CO2: Analyse images in the frequency domain using various transforms. (Level: Analyse)

CO3: Evaluate the techniques for image enhancement and image restoration. (Level: Evaluate)

CO4: Categorize various compression techniques. (Level: Analyse)

CO5: Interpret Image compression standards. (Level: Evaluate)

CO6: Interpret image segmentation and representation techniques. (Level: Evaluate)

Detailed Syllabus:

Module	Topics	No of Lectures
1	<p>Introduction to Image-processing System: Introduction, Image Sampling, Quantization, Resolution, Human Visual Systems, Elements of an Image-processing System, Applications of Digital Image Processing</p> <p>2D Signals and Systems: 2D signals, separable sequence, periodic sequence, 2D systems, classification of 2D systems, 2D Digital filter</p> <p>Convolution and Correlation: 2D Convolution through graphical method, Convolution through 2D Z—transform, 2D Convolution through matrix analysis, Circular Convolution, Applications of Circular Convolution, 2DCorrelation</p>	9
2	<p>Image Transforms: Need for transform, image transforms, Fourier transform, 2D Discrete Fourier Transform, Properties of 2D DFT</p> <p>Image Enhancement: Image Enhancement in spatial domain, Enhancement through Point operations, Histogram manipulation, Linear and nonlinear Gray Level Transformation, local or neighborhood operation, Median Filter, Spatial domain High pass filtering, Bit-plane slicing, Image Enhancement in frequency domain, Homomorphic filter, Zooming operation, Image Arithmetic</p>	9
3	<p>Binary Image processing: Mathematical morphology, Structuring elements, Morphological image processing, Logical operations, Morphological operations, Dilation and Erosion, Distance Transform</p> <p>Colour Image processing: Colour images, Colour Model, Colour image quantization, Histogram of a colour image</p>	9
4	<p>Fundamentals, Huffman Coding, Golomb Coding, Arithmetic Coding, LZW Coding, Run-length Coding, Symbol-based Coding, 8 Bit-plane Coding, Block Transform Coding, Predictive Coding, Wavelet Coding, Digital Image Watermarking</p>	9

5	Image segmentation techniques, Region approach, Clustering techniques, Thresholding, Edge-based segmentation, Edge detection, Edge Linking, Hough Transform	9
---	---	---

Reference Books

1. Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar, Tata McGraw-Hill Education Pvt. Ltd., 2009
2. Digital Image Processing 3rd Edition, Rafael C Gonzalez, Richard E Woods, Pearson, 2008
3. Scilab Textbook Companion for Digital Image Processing, S. Jayaraman, S. Esakkirajan And T. Veerakumar, 2016 (https://scilab.in/textbook_companion/generate_book/125)

Practical:

1. Perform Linear and Circular Convolution between two matrices
2. Apply DFT on an image
3. Apply the following Pre-Processing Techniques on an Image: Log Transform, Power Law Transform, Image Negation
4. Apply the following Image Enhancement Techniques on an Image: Brightness Adjustment, Contrast Stretching, Thresholding, Gray Level Slicing
5. Color Image Processing – I: Splitting RGB Planes, Pseudo Coloring
6. Color Image Processing – II: Brightness Adjustment, Contrast Stretching, Thresholding, Gray Level Slicing
7. Write a program to perform Histogram Equalization of an image
8. Write a program to perform Smoothing on an image
9. Write a program to perform Sharpening on an image
10. Write a program to perform Dilation and Erosion on an image.

Project

Semester V & VI

(Implemented during Academic Year 2022-23)

Course Objective:

- To provide an opportunity to apply the knowledge gained through various courses in solving a real life problem.
- To provide an opportunity to practice different phases of software/system development life cycle.
- To introduce the student to a professional environment and/or style typical of a global IT industry.
- To provide an opportunity for structured team work and project management.
- To provide an opportunity for effective, real-life, technical documentation.
- To provide an opportunity to practice time, resource and person management.

Course Outcome:

- Student will have exposure to understand industry-standard project practices, through a real-life project work under time and deliverable constraints, applying the knowledge acquired through various courses.(Apply,Create)

The project dissertation/document is expected to be created and it should have the following contents.

- a. SRS – Problem Statement, BRD- Business Requirement Document
- b. General Requirement
- c. System design (RED/Class Diagrams, DFD/Activity diagrams/Circuit diagram)
- d. User screen design and client side validation
- e. Database Design
- f. User interface design /user manual
- g. Test cases
- h. Future Scope and limitation
- i. Conclusion
- j. Bibliography

Evaluation Scheme

Internal Exam-25 Marks

- **Test– 20 Marks which will be converted out of 10 Marks-** Duration 40 mins

It will be conducted either as a written test or using any open source learning management system such as Moodle (Modular Object-Oriented Dynamic Learning Environment) or a test based on an equivalent online course on the contents of the concerned course (subject) offered by or build using MOOC (Massive Open Online Course) platform.

- **Assignments/Presentation/Projects – 10 Marks**

Subject specific Term Work Module/assessment modes –as decided by the department in the beginning of the semester (like Extension/field/experimental work, Short Quiz; Objective test, lab practical, open book test etc. and written assignments, Case study, Projects, Posters and exhibits etc. for which the assessment is to be based on class presentations wherever applicable)

- **Active participation in routine class instructional deliveries - 5 Marks**

Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

External Examination- 75 Marks

Duration - 2.5 Hours.

Theory question paper pattern,-

All questions are compulsory.		
Question	Based on	Marks
Q.1	Unit 1	15
Q.2	Unit 2	15
Q.3	Unit 3	15
Q.4	Unit 4	15
Q. 5	Unit 5	15

All questions shall be compulsory with internal choice within the questions.

Each Question may be sub-divided into sub questions as a, b, c, d & e, etc & the allocation of Marks depends on the weightage of the topic.

Practical Examination – 50 marks (Duration, 2 Hours)

- Each practical course carries 50 Marks , 40 marks + 05 marks (journal)+ 05 marks(viva)
- Minimum 75% practical from each core/allied course are required to be completed and written in the journal.

(Certified Journal is compulsory for appearing at the time of Practical Exam)