# Nadindas Khandwala College

**Revised Syllabus**
**And**
**Question Paper Pattern**
**Of Course**
**Of**
**Bachelor of Computer Science (BSC CS)**
**Programme**

**Department Of CS**
**First Year**
***Semester I***

**Under Autonomy**

***(To be implemented from Academic Year- 2017-2018)***

# Bachelor of Computer Science (BSC CS) Program

*Under Choice Based Credit, Grading and Semester System*

*Course Structure*

## FYBSC CS

*(To be implemented from Academic Year- 2017-2018)*

| FYBSC CS – SEMESTER I | | | | | | | |
|---|---|---|---|---|---|---|---|
| Course Code | Course | Hrs. of Instruction/Week | Exam Duration (Hours) | Maximum Marks | | | Credits |
| | | | | CIE | SEE | Total | |
| 1711UCSSS | **Part 1** Soft Skills Development | 2 | 2 1/2 Hours | 25 | 75 | 100 | 2 |
| 1712UCSCO | **Core 1:** Computer Organization and Design | 3 | 2 1/2 Hours | 25 | 75 | 100 | 2 |
| 1713UCSIP | **Core 2:** Introduction to Programming | 3 | 2 1/2 Hours | 25 | 75 | 100 | 2 |
| 1714UCSFS | **Core 3:** Free and Open Source Software | 3 | 2 1/2 Hours | 25 | 75 | 100 | 2 |
| 1715UCSDB | **Core 4:** Database Systems | 3 | 2 1/2 Hours | 25 | 75 | 100 | 2 |
| 1716UCSDM | **Core 5:** Discrete Mathematics | 3 | 2 1/2 Hours | 25 | 75 | 100 | 2 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **1717UCSSP** | **DSE Allied 1 :**<br>Descriptive Statistics and Introduction to Probability | **3** | **2 1/2 Hours** | **25** | **75** | **100** | **2** |
| **1712UCSPR** | **Core 1 Practical:**<br>Computer Organization and Design | **2** | **2 Hours** | **-** | **50** | **50** | **1** |
| **1713UCSPR** | **Core 2 Practical:**<br>Introduction to Programming | **2** | **2 Hours** | **-** | **50** | **50** | **1** |
| **1714UCSPR** | **Core 3 Practical:**<br>Free and Open Source Software | **2** | **2 Hours** | **-** | **50** | **50** | **1** |
| **1715UCSPR** | **Core 4 Practical:**<br>Database Systems | **2** | **2 Hours** | **-** | **50** | **50** | **1** |
| **1716UCSPR** | **Core 5 Practical:**<br>Discrete Mathematics | **2** | **2 Hours** | **-** | **50** | **50** | **1** |
| **1717UCSPR** | **DSE Allied 1 Practical:**<br>Descriptive Statistics and Introduction to Probability | **2** | **2 Hours** | **-** | **50** | **50** | **1** |
| | **TOTAL** | **32** | | | | | **20** |

| Course Code | Course | Hrs. of Instruction/ week | Exam Duration (Hours) | Maximum Marks | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | CIE | SEE | Total | |
| 1711UCSSS | **Part 1:** **Soft Skills Development** | 3 | 2 ½ hrs | 25 | 75 | 100 | 2 |

| Sr. No. | Modules / Units |
|---|---|
| 1 | **UNIT 1** |
| | **Personality Development:** Knowing Yourself, Positive Thinking, Johari's Window, verbal Communication, Non-verbal Communication, Body Language: grooming<br><br>**Emotional Intelligence:** Meaning and Definition, Need for Emotional Intelligence, Intelligence Quotient versus Emotional Intelligence Quotient, Components of Emotional Intelligence, Competencies of Emotional Intelligence, Skills to Develop Emotional Intelligence<br><br>**Etiquette and Mannerism:** Introduction, Professional Etiquette, Technology Etiquette<br><br>**Communication Today:** Significance of Communication, GSC's 3M Model of Communication, Vitality of the Communication Process, Fundamentals of Good Listening, Need for Intercultural Communication, Communicating Digital World |
| 2 | **UNIT 2** |
| | **Academic Skills**<br><br>**Resume writing:** Introduction, Resume, Curriculum Vitae, Job Application or Cover Letter<br><br>**Professional Presentation:** Planning a Presentation, Preparing the Presentation, Delivering the Presentation<br><br>**Job Interviews:** Types of job Interviews, Preparatory Steps for Job Interviews, Interview Skill Tips, FAQ During Interviews<br><br>**Group Discussion:** Difference between Group Discussion, Panel Discussion and Debate, Importance of Group Discussions, Traits, Types of Group Discussions, Individual Traits |

| 3 | UNIT 3 |
|---|---|
| | **Creativity at Workplace:** Current Workplaces, Nurturing Hobbies at Work, The Six Thinking Hat Method<br><br>**Ethical Values:** Ethics and Society, Correlation between Values and Behavior, Nurturing Ethics, Importance of Work Ethics, Problems in the Absence of Work Ethics<br><br>**Leadership and Team Building:** Leader and Leadership, Leadership Traits, Culture and<br><br>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<br><br>**Stress and Time Management:** Stress, Sources of Stress, Ways to Cope with Stress, time management, prioritising and procrastination |

| Reference Books |
|---|
| **Soft Skills Development** |
| **Text book:**<br><br>1. Soft Skills: an Integrated Approach to Maximise Personality, Gajendra S. Chauhan, Sangeeta Sharma, Wiley India<br><br>**Additional References:**<br><br>1. Personality Development and Soft Skills, Barun K. Mitra, Oxford Press<br>2. Business Communication, Shalini Kalia, Shailja Agrawal, Wiley India<br>3. Soft Skills - Enhancing Employability, M. S. Rao, I. K. International<br>4. Cornerstone: Developing Soft Skills, Sherfield, Pearson India |

| Course Code: | Course | Hrs. of Instruction/ week | Exam Duration (Hours) | Maximum Marks | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | CIE | SEE | Total | |
| 1712UCSCO | Computer Organization and Design | 3 | 2 ½ hrs | 25 | 75 | 100 | 2 |

| Sr. No. | Modules / Units |
|---|---|
| 1 | **UNIT 1** |
| | **Computer Abstractions and Technology**: Basic structure and operation of a computer, functional units and their interaction. Representation of numbers and characters.<br>**Logic circuits and functions:** Combinational circuits and functions: Basic logic gates and functions, truth tables; logic circuits and functions. Minimization with Karnaugh maps. Synthesis of logic functions with and-or-not gates, nand gates, nor gates. Fan-in and fan-out requirements; tristate buffers. Half adder, full adder, Half subtractor, full subtractor ripple carry adder. Shift registers, Decoders and multiplexers registers.<br>**Sequential circuits and functions:** State diagram and state table; finite state machines and their synthesis. |
| 2 | **UNIT 2** |
| | **Instruction set architectures**:<br>Memory organization, addressing and operations; word size, big-endian and little- endian arrangements. Instructions, sequencing. Instruction sets for RISC and CISC (examples Altera NIOS II and Freescale ColdFire). Operand addressing modes; pointers; indexing for arrays. Machine language, assembly language, assembler directives. Function calls, processor runtime stack, stack frame. Types of machine instructions: arithmetic, logic, shift, etc. Instruction sets. |
| 3 | **UNIT 3** |
| | **Basic Processor Unit:**<br>Main components of a processor: registers and register files, ALU, control unit, instruction fetch unit, interfaces to instruction and data memories. Datapath. Instruction fetch and execute; executing arithmetic/logic, memory access and branch instructions; hardwired and microprogrammed control for RISC and CISC.<br>**Basic I/O:**<br>Accessing I/O devices, data transfers between processor and I/O devices.<br>Interrupts and exceptions: interrupt requests and processing. |

| Reference Books |
| --- |
| Computer Organization and Design |
| **Text book:**<br>1. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012<br><br>**Additional References:**<br>1. Patterson and Hennessy, Computer Organization and Design, Morgan Kaufmann, ARM Edition, 2011<br>R P Jain, Modern Digital Electronics, Tata McGraw Hill Education Pvt. Ltd. , 4th Edition, 2010 |

| **Practical (**1712UCSPR) |
| --- |
| 1) Study and verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, and EX-NOR).<br>2) Simplify given Boolean expression and realize it.<br>3) Design and verify a half/full adder<br>4) Design and verify half/full subtractor<br>5) Design a 4 bit magnitude comparator using combinational circuits.<br>6) Design and verify the operation of flip-flops using logic gates.<br>7) Verify the operation of a counter.<br>8) Verify the operation of a 4 bit shift register<br>9) Using SPIM, write and test an adding machine program that repeatedly readsin integers and adds them into a running sum. The program should stop whenit gets an input that is 0, printing out the sum at that point.<br>10) Using SPIM, write and test a program that reads in a positive integer using theSPIM system calls. If the integer is not positive, the program should terminatewith the message “Invalid Entry”; otherwise the program should print out thenames of the digits of the integers, delimited by exactly one space. For example, if the user entered “528,” the output would be “Five Two Eight.” |

| Course Code | Course | Hrs. of Instruction/Week | Exam Duration (Hours) | Maximum Marks | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | CIE | SEE | Total | |
| 1713UCSIP | Core 1: Introduction to Programming | 3 | 2 1/2 Hours | 25 | 75 | 100 | 2 |

| Sr.No. | Modules / Units |
|---|---|
| 1 | **UNIT 1** |
| | Strategies for problem solving:<br>Classic Puzzles: How to cross the river with fox, goose and corn, The sliding eight, sliding five, completing a Sudoku square, Quarrasi lock(Implementation through scratch)<br>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 |
| 2 | **UNIT 2** |
| | Output method, Output patterns: Half of a square, count down by counting up, sideways triangle, Input processing: convert character digit to integer, checksum validation, positive or negative, decode a message(Implementation using Python) |
| 3 | **UNIT 3** |
| | List: store, copy, search, sort, find mean/median/mode<br>Function: introduction, function definition, parameters, function call<br>Recursion: parrot counting, finding the highest revenue customer, iterative sum of list elements |

| Reference Books |
| --- |
| **Introduction to Programming** |

**Text book:**


1. V. Anton Spraul, Think like a Programmer: An Introduction to Creative problem Solving, No Starch press Inc, 2012
2. Magnus Lie Hetland, Beginning Python: From Novice to Professional, Apress
3. Paul Gries, Practical Programming: An Introduction to Computer Science Using Python Pragmatic Bookeshelf, 2nd Edition, 2014


**Additional References:**


1. Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers, How to Think Like a Computer Scientist: Learning with Python 3 Documentation, Release 3rd Edition, 2012
2. David Evans, Introduction to Computing: Explorations in Language, Logic, and Machines, Creative Commons, 2011


| **Practical (**1713UCSPR) |
| --- |

1) Write a Scratch program for (any one):
   a) How to cross the river with fox, goose and corn
   b) The sliding eight
   c) sliding five
2) Write Python program for printing the patterns
3) Write Python program to check:
   a) Whether the given number is odd or even
   b) whether the given year is a leap year
4) Write Python program to convert:
   a) character digit to integer
   b) decimal number to binary or hexadecimal
5) Write 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 Python program to find:
   a) The nth Fibonacci number
   b) Factorial of a given number
7) Write Python program to convert read a line of text and count
   a) the number of letters
   b) number of words
8) Write Python program to read a list of values and determines the quartiles
9) Write a Python function to check whether the list is sorted in one pass.
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) binary search
   b) tower of Hanoi

| Course Code | Course | Hrs. of Instruction/Week | Exam Duration (Hours) | Maximum Marks | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | CIE | SEE | Total | |
| 1714UCSFS | Core 3: Free and Open Source Software | 3 | 2 1/2 Hours | 25 | 75 | 100 | 2 |

| Sr.No. | Modules / Units |
|---|---|
| 1 | **UNIT 1** |
| | **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.<br><br>**Methodologies:** Open Source History, Initiatives, Principle and methodologies. Philosophy : Software Freedom, Open Source Development Model Licenses and Patents: What Is A License, Important FOSS Licenses (Apache,BSD,GPL, LGPL), copyrights and copy lefts, Patents Economics of FOSS : Zero Marginal Cost, Income-generation opportunities, Problems with traditional commercial software, Internationalization<br><br>**Social Impact:** Open source vs. closed source, Open source government, Open source ethics. Social and Financial impacts of open source technology, Shared software, Shared source, Open Source in Government. |
| 2 | **UNIT 2** |
| | **Case Studies**<br><br>Example Projects: Apache web server, GNU/Linux, Android, Mozilla (Firefox), Wikipedia, Drupal, wordpress, GCC, GDB, github, Open Office. Study: Understanding the developmental models, licensings, mode of funding,commercial/non-commercial use. Open Source Hardware, Open Source Design, Open source Teaching. Open source media.<br><br>**Collaboration, Community and Communication Contributing to Open Source Projects:** Introduction to github, interacting with the community on github, Communication and etiquette, testing open source code, reporting issues, contributing code. Introduction to wikipedia, contributing to Wikipedia Or contributing to any prominent open source project of student's choice. Starting and Maintaining own Open Source Project. |
| 3 | **UNIT 3** |
| | **Understanding Open Source Ecosystem:** Open Source Operating Systems: GNU/Linux, Android, Free BSD, Open Solaris. Open Source Hardware, Virtualization Technologies, Containerization Technologies: Docker, Development tools, IDEs, debuggers, Programming languages, LAMP, Open Source database technologies |

| Reference Books |
| --- |
| **Free and Open Source Software** |

**Text book:**

1. Unix Concepts and Applications by Sumitabha Das, Tata McGraw Hill Education, 2006
2. The official Ubuntu Book, 8th Edition

**Additional References:**

1. The Linux Documentation Project: http://www.tldp.org/
2. Docker Project Home: http://www.docker.com
3. Linux kernel Home: http://kernel.org
4. Open Source Initiative: https://opensource.org/
5. Linux Documentation Project: http://www.tldp.org/
6. Wikipedia: https://en.wikipedia.org/
7. https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia
8. Github: https://help.github.com/

The Linux Foundation: http://www.linuxfoundation.org/

| **Practical (**1714UCSPR) |
| --- |

**Free and Open Source Software**

**1. Identify any Open Source software and create detailed report about it.**

Sample Guidelines.

a. Idea
b. What problem does it solves?
c. Licensing model
d. Intent behind making it open source
e. Monetization models
f. Popularity
g. Impact

**2. Learn at least three different open source licenses and create a brief report** about them.

a. History of license
b. Idea
c. What problems does it solve?
d. Detailed licensing model
e. Which popular software are released under this license?
f. Any popular news associated with this license?
g. Popularity
h. Impact

**3. Contributing to Open Source**

a. Identify any Open Source project of your interest
b. Learn more about the project w.r.t. Lab 1.
c. Start contributing to the project either by
i. Testing
ii. Reporting bugs
Iii. Coding
iv. Helping in documentation
v. Participating in discussions
vi. Participating in pre-release testing programs
vii. UI development.
viii. Or any other important area.

**4. Hands on with Open Source Software**

a. Identify any open source software of your interest
b. Learn it from practical view-point
c. Give a brief presentation about it to the class
d. Sample projects: gcc, gdb, drupal, wordpress, apache web server, mysql database

**5. Contributing to Wikipedia:**

a. Introduction to wikipedia: operating model, license, how to contribute?
b. Create your user account on wikipedia
c. Identify any topic of your choice and contribute the missing information

**6. Github**

a. Create and publish your own open source project: Write any simple program using your choice of programming language.
b. Create a repository on github and save versions of your project. You'll learn about the staging area, committing your code, branching, and merging,
c. Using GitHub to Collaborate: Get practice using GitHub or other remote repositories to share your changes with others and collaborate on multi-developer projects. You'll learn how to make and review a pull request on GitHub.
d. Contribute to a Live Project: Students will publish a repository containing their reflections from the course and submit a pull request.

**7. Open Source Operating Systems**

a. Learn any open source operating system of your choice : Linux, Android, FreeBSD, Open Solaris etc.
b. Learn the installation.
c. Identify the unique features of the OS of your choice.
d. Learn linux commands:
pwd, cd, absolute and relative paths, ls, mkdir, rmdir,
file, touch, rm, cp. mv, rename, head, tail, cat, tac, chmod

**8. Virtualization: Open Source virtualization technologies:**

a. Install and configure any one: VirtualBox, Zen, KVM
b. Create and use virtual machines

**9. Containerization:**

a. Containerization technologies: docker, rocket, LXD
b. Install and configure any containerization technology
c. Create and use containers using it

**10. Linux Kernel: Learn Linux kernel with respect to:**

a. What is Linux kernel?
b. Operating model
c. Licensing Model
d. How development works?
e. Download kernel source code.
f. Compile the Kernel

| Course Code: | Course | Hrs. of Instruction/ week | Exam Duration (Hours) | Maximum Marks | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | CIE | SEE | Total | |
| 1715UCSDB | DSE 1: Database Systems | 3 | 2 ½ hrs | 25 | 75 | 100 | 2 |

| Sr. No. | Modules / Units |
|---|---|
| 1 | **UNIT 1** |
| | **Introduction to DBMS –** Database, DBMS – Definition, Overview of DBMS, Advantages of DBMS, Purpose of Database Systems, View of Data, Database Languages, Relational Databases, Database Design – Data Storage and Querying, Levels of abstraction, Data independence, DBMS Architecture<br><br>Data models - Client/Server Architecture, Object Based Logical Model, Record Based Logical Model ( relational, hierarchical, network)<br><br>**Transaction management and Concurrency control -** Transaction management: ACID properties, serializability and concurrency control, Lock based concurrency control (2PL, Deadlocks),Time stamping methods, database recovery management<br><br>**Entity Relationship Model -** Entities, attributes, entity sets, relations, relationship sets, Additional constraints ( key constraints, participation constraints, weak entities, aggregation / generalization, Conceptual Design using ER ( entities VS attributes, Entity Vs relationship, binary Vs ternary, constraints beyond ER)<br><br>**Relational data model–** Domains, attributes, Tuples and Relations, Relational Model Notation, Characteristics of Relations, Relational Constraints - primary key, referential integrity, unique constraint, Null constraint, Check constraint<br><br>**ER to Table**- Entity to Table, Relationship to tables with and without key constraints. |

| | |
|---|---|
| **2** | **UNIT 2** |
| | **Schema refinement and Normal forms:** Functional dependencies, first, second, third, and BCNF normal forms based on primary keys, lossless join decomposition.<br><br>**Relational Algebra:** operations (selection, projection, set operations union, intersection, difference, cross product, Joins –conditional, equi join and natural joins, division)<br><br>**DDL Statements** - Creating Databases, Using Databases, data types, Creating Tables (with integrity constraints – primary key, default, not null, check, foreign key), Altering Tables, Renaming Tables, Dropping Tables, Truncating Tables, Backing Up and Restoring databases<br><br>**DML Statements** – Viewing the structure of a table insert, update, delete, Select all columns, specific columns, unique records, conditional select, in clause, between clause, limit, aggregate functions (count, min, max, avg, sum), group by clause, having clause, Pattern matching |
| **3** | **UNIT 3** |
| | **Functions** – String Functions (concat, instr, left, right, mid, length, lcase/lower, ucase/upper, replace, strcmp, trim, ltrim, rtrim), Math Functions (abs, ceil, floor, mod, pow, sqrt, round, truncate) Date Functions (adddate, datediff, day, month, year, hour, min, sec, now, reverse)<br><br>**Joining Tables** – inner join, outer join (left outer, right outer, full outer)<br><br>**Subqueries** – subqueries with IN, EXISTS, subqueries restrictions, Nested subqueries, ANY/ALL clause, correlated subqueries<br><br>**Database Protection:** Security Issues, Threats to Databases, Security Mechanisms, Role of DBA, Discretionary Access Control<br><br>**Views -** creating, altering dropping, renaming and manipulating views<br><br>**DCL Statements -** creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges |

| **Reference Books** |
|---|
| **Introduction to Financial Accounts** |
| **Text books:**<br>1. Ramez Elmasri & Shamkant B.Navathe, Fundamentals of Database Systems, Pearson Education, Sixth Edition, 2010<br>2. Ramakrishnam, Gehrke, Database Management Systems, McGraw-Hill, 2007<br>3. Joel Murach, Murach's MySQL, Murach, 2012<br><br>**Additional References:**<br>Robert Sheldon, Geoff Moes, Begning MySQL, Wrox Press, 2005. |

**Practical (**1715UCSPR)

1. Database Systems
   For given scenario. Draw ER diagram and convert entities and relationships to table.

2. Write relational algebra queries on the tables created in Practical
3. Perform the following
   - Viewing all databases
   - Creating a Database
   - Viewing all Tables in a Database
   - Creating Tables (With and Without Constraints)
   - Inserting/Updating/Deleting Records in a Table
   - Saving (Commit) and Undoing (rollback)
4. Perform the following:
   - Altering a Table
   - Dropping/Truncating/Renaming Tables
   - Backing up / Restoring a Database
5. Perform the following:
   - Simple Queries
   - Simple Queries with Aggregate functions
   - Queries with Aggregate functions (group by and having clause)
6. Queries involving
   - Date Function
   - String Function
   - Math Functions
7. Join Queries
   - Inner Join
   - Outer Join(left,right,full)
8. Subqueries
   - With IN clause
   - With EXISTS clause
9. Views
   - Creating Views (with and without check option)
   - Dropping views
   - Selecting from a view
10. DCL statements
    - Granting and revoking permissions

| Course Code | Course | Hrs. of Instruction/Week | Exam Duration (Hours) | Maximum Marks | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | CIE | SEE | Total | |
| 1616UCSDM | DSE-Core 2: Discrete Mathematics | 3 | 2 1/2 Hours | 25 | 75 | 100 | 2 |

| Sr.No. | Modules / Units |
|---|---|
| 1 | **UNIT 1** |
| | **Recurrence Relations**<br>a) **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.<br>b) **Relations**: Definition and examples. Properties of relations , Partial Ordering sets, Linear Ordering Hasse Diagrams , Maximum and Minimum elements, Lattices<br>**Recurrence Relations:** Definition of recurrence relations, Formulating recurrence relations, solving recurrence relations- Back tracking method, Linear homogeneous recurrence relations with constant coefficients. Solving linear homogeneous recurrence relations with constant coefficients of degree two when characteristic equation has distinct roots and only one root, Particular solutions of non linear homogeneous recurrence relation, Solution of recurrence relation by the method of generation functions, Applications- Formulate and solve recurrence relation for Fibonacci numbers, Tower of Hanoi, Intersection of lines in a plane, Sorting Algorithms. |
| 2 | **UNIT 2** |
| | **Counting Principles , Languages and Finite State Machine**<br>a) **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.<br>b) **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)<br>c) **Languages, Grammars and Machines:** Languages , regular Expression and Regular languages, Finite state Automata, grammars, Finite state machines, Gödel numbers, Turing machines. |

| 3 | UNIT 3 |
|---|---|
| | **Graphs and Trees**<br><br>a) **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.<br><br>**b) 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 |

**Reference Books**

**Discrete Mathematics**

**Textbook**:

1. Discrete Mathematics and Its Applications, Seventh Edition by Kenneth H. Rosen, McGraw Hill Education (India) Private Limited. (2011)
2. Norman L. Biggs, Discrete Mathematics, Revised Edition, Clarendon Press, Oxford 1989.
3. Data Structures Seymour Lipschutz, Schaum's out lines, McGraw- Hill Inc.

**Additional References**:

1. Elements of Discrete Mathematics: C.L. Liu , Tata McGraw- Hill Edition .
2. Concrete Mathematics (Foundation for Computer Science): Graham, Knuth, Patashnik Second Edition, Pearson Education.
3. Discrete Mathematics: Semyour Lipschutz, Marc Lipson, Schaum's out lines, McGraw-Hill Inc.
4. Foundations in Discrete Mathematics: K.D. Joshi, New Age Publication, New Delhi.

| **Practical (**1716UCSPR) |
| --- |
| 1. Graphs of standard functions such as absolute value function, inverse function, logarithmic and exponential functions, flooring and ceiling functions, trigonometric functions over suitable intervals.<br>2. Partial ordering sets, Hasse diagram and Lattices.<br>3. Recurrence relation.<br>4. Different counting principles.<br>5. Finite state Automata and Finite state machines.<br>6. Warshall's Algorithm.<br>7. Shortest Path algorithms.<br>8. Operations on graph.<br>9. Breadth and Depth First search algorithms.<br>10. Concept of searching, inserting and deleting from binary search trees. |

| Course Code | Course | Hrs. of Instruction/Week | Exam Duration (Hours) | Maximum Marks | | | Credits |
|---|---|---|---|---|---|---|---|
| | | | | CIE | SEE | Total | |
| **1617UCSSP** | **DSE-Allied 1:** Descriptive Statistics and Introduction to Probability | **3** | **2 1/2 Hours** | **25** | **75** | **100** | **2** |

| Sr.No. | Modules / Units |
|---|---|
| **1** | **UNIT 1** |
| | **Data Presentation:**Data types : attribute, variable, discrete and continuous variable<br>Data presentation : frequency distribution, histogram o give, curves, stem and leaf display<br><br>**Data Aggregation:**Measures of Central tendency: Mean, Median, mode for raw data, discrete, grouped frequency distribution.<br><br>Measures dispersion: Variance, standard deviation, coefficient of variation for raw data, discrete and grouped frequency distribution, quartiles, quantiles Real life examples |
| **2** | **UNIT 2** |
| | **Moments:** raw moments, central moments, relation between raw and central moments<br>**Measures of Skewness and Kurtosis:** based on moments, quartiles, relation between mean, median, mode for symmetric, asymmetric frequency curve.<br>**Correlation and Regression:** bivariate data, scatter plot, correlation, nonsense correlation, Karl pearson's coefficients of correlation, independence.<br>**Linear regression:** fitting of linear regression using least square regression, coefficient of determination, properties of regression coefficients (only statement) |
| **3** | **UNIT 3** |
| | **Probability :** Random experiment, sample space, events types and operations of events<br><br>**Probability definition :** classical, axiomatic, Elementary Theorems of probability (without proof)<br>$0 \leq P(A) \leq 1$,<br><br>$P(A \cup B) = P(A) + P(B) - P(A \cap B)$<br><br>$P(A') = 1 - P(A)$<br><br>$P(A) \leq P(B)$ if A C B<br><br>Conditional probability, 'Bayes' theorem, independence, Examples on Probability |

## Reference Books

Descriptive Statistics and Introduction to Probability

**Text Book:**


1. Trivedi, K.S.(2001) : Probability, Statistics, Design of Experiments and Queuing theory, with applications of Computer Science, Prentice Hall of India, New Delhi


**Additional References:**


1. Ross, S.M. (2006): A First course in probability. 6th Edn Pearson
2. Kulkarni, M.B., Ghatpande, S.B. and Gore, S.D. (1999): common statistical tests. Satyajeet Prakashan, Pune
3. Gupta, S.C. and Kapoor, V.K. (1987): Fundamentals of Mathematical Statistics, S. Chand and Sons, New Delhi
4. Gupta, S.C. and Kapoor, V.K. (1999): Applied Statistics, S. Chand and Son's, New Delhi

Montgomery, D.C. (2001): Planning and Analysis of Experiments, wiley.


**Practical (**1717UCSPR)

1. Frequency distribution and data presentation
2. Measures of central tendency
3. Data entry using, functions, c(), scan (), Creating vectors, Mathematical Operations: ** +/-/*/ / ^ , exp, log, log10, etc, creating vector of text type, useful functions: data, frame, matrix operations, seq(), split() etc.
4. Frequency distribution using cut(), table()
5. Data presentation
6. Summary Statistics (measures of central tendency, dispersion)
7. Measures of skewness and kurtosis
8. Correlation and regression
9. Probability
10. Conditional probability

# Evaluation Scheme

**I. Internal Exam-25 Marks**

**(i) Test– 20 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.

**(ii) 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.

**II. External Examination- 75 Marks**

**(i)** Duration - 2.5 Hours.

**(ii)** Theory question paper pattern:-

| All questions are compulsory. | | |
|---|---|---|
| Question | Based on | Marks |
| Q.1 | Unit I | 20 |
| Q.2 | Unit II | 20 |
| Q.3 | Unit III | 20 |
| Q.4 | Unit I,II and III | 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.

**III. 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)**