Nagindas Khandwala College

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

(Department Of CS) Second Year

Semester III

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

SYBSC CS

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

SYBSC CS – SEMESTER III							
Course Code	Course	Hrs. of Instruction /Week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1731UCSTC	Core Subject: Theory of Computation	3	2 $^{1/2}$ Hours	25	75	100	2
1732UCSCJ	Core Subject: Core Java	3	2 $^{1/2}$ Hours	25	75	100	2
1733UCSOS	Core Subject: Operating System	3	2 $^{1/2}$ Hours	25	75	100	2
1734UCSDB	Core Subject: Database Management System	3	2 $^{1/2}$ Hours	25	75	100	2
1735UCSCG	Core Subject: Combinatorics and Graph Theory	3	2 $^{1/2}$ Hours	25	75	100	2
1736UCSPC	Core Subject: Physical Computing Programming and IoT	3	2 ½ Hours	25	75	100	2
1737UCSWP	Skill Enhancement:	3	2 ½ Hours	25	75	100	2

	Web Programming						
1732UCSPR	**Core Subject Practical 2:** **Core Java**	**2**	**2 Hours**		**50**	**50**	**1**
1733UCSPR	**Core Subject Practical 3:** **Operating System**	**2**	**2 Hours**		**50**	**50**	**1**
1734UCSPR	**Core Subject Practical 4:** **Database Management System**	**2**	**2 Hours**		**50**	**50**	**1**
1735UCSPR	**Core Subject Practical 5:** **Combinatorics and Graph Theory**	**2**	**2 Hours**		**50**	**50**	**1**
1736UCSPR	**Core Subject Practical 6:** **Physical Computing & IoT Programming**	**2**	**2Hours**		**50**	**50**	**1**
1737UCSPR	**Skill Enhancement:** **Practical 7: Web Programming**	**2**	**2 Hours**		**50**	**50**	**1**
	TOTAL	**33**					**20**

Course Code	Course	Hrs. of Instruction/ week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1731UCSTC	Core Subject: Theory of Computation	3	2 $^{1/2}$ Hours	25	75	100	2

Sr. No.	Modules / Units
1	UNIT 1
	Automata Theory: Defining Automaton, Finite Automaton, Transitios and Its properties, Acceptability by Finite Automaton, Nondeterministic Finite State Machines, DFA and NDFA equivalence, Mealy and Moore Machines, Minimizing Automata. **Formal Languges:** Defining Grammar, Derivations, Languges generated by Grammar, Comsky Classification of Grammar and Languages, Recursive Enumerable Sets, Operations on Languages, Languages and Automata
2	UNIT 2
	Regular Sets and Regular Grammar: Regular Grammar, Regular Expressions, Finite automata and Regular Expressions, Pumping Lemma and its Applications, Closure Properties, Regular Sets and Regular Grammar **Context Free Languages:** Context-free Languages, Derivation Tree, Ambiguity of Grammar, CFG simplification, Normal Forms, Pumping Lemma for CFG **Pushdown Automata:** Definitions, Acceptance by PDA, PDA and CFG
3	UNIT 3
	Linear Bound Automata: The Linear Bound Automata Model, Linear Bound Automata and Languages. **Turing Machines:** Turing Machine Definition, Representations, Acceptability by Turing Machines, Designing and Description of Turing Machines, Turing Machine Construction, Variants of Turing Machine, **Undecidability:** The Church-Turing thesis, Universal Turing Machine, Halting Problem, Introduction to Unsolvable Problems

Reference Books

Theory of Computing

Reference book:

1) Theory of Computer Science, K. L. P Mishra, Chandrasekharan, PHI,3rd Edition
2) Introduction to Computer Theory, Daniel Cohen, Wiley,2nd Edition
3) Introductory Theory of Computer Science, E.V. Krishnamurthy,Affiliated East-West Press.

Additional Reference(s):
1) Theory of Computation, Kavi Mahesh, Wiley India
2) Elements of The Theory of Computation, Lewis, Papadimitriou, PHI
3) Introduction to Languages and the Theory of Computation, John E Martin, McGraw-Hill Education
4) Introduction to Theory of Computation, Michel Sipser, Thomson

Tutorial (Theory of Computing)

Tutorials :
1. Problems on generating languages for given simple grammar
2. Problems on DFA and NDFA equivalence
3. Problems on generating Regular Expressions
4. Problems on drawing transition state diagrams for Regular Expressions
5. Problems on Regular Sets and Regular Grammar
6. Problems on Ambiguity of Grammar
7. Problems on working with PDA
8. Problems on working with Turing Machines
9. Problems on generating derivation trees
10. Problems on Linear Bound Automata/Universal Turing Machine

Course Code:	Course	Hrs. of Instruction/ week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1732UCSCJ	Core Subject: Core Java	3	2 $^{1/2}$ Hours	25	75	100	2

Sr. No.	Modules / Units
1	UNIT 1
	The Java Language: Features of Java, Java programming format, Java Tokens, Java Statements, Java Data Types, Typecasting, Arrays **OOPS**: Introduction, Class, Object, Static Keywords, Constructors, this Key Word, Inheritance, super Key Word, Polymorphism (overloading and overriding), Abstraction, Encapsulation, Abstract Classes, Interfaces **String Manipulations**: String, String Buffer, String Tokenizer **Packages:** Introduction to predefined packages (java.lang, java.util, java.io, java.sql, java.swing), User Defined Packages, Access specifiers
2	UNIT 2
	Exception Handling: Introduction, Pre-Defined Exceptions, Try-Catch-Finally, Throws, throw, User Defined Exception examples **Multithreading:** Thread Creations, Thread Life Cycle, Life Cycle Methods, Synchronization, Wait() notify() notify all() methods **I/O Streams:** Introduction, Byte-oriented streams, Character- oriented streams, File, Random access File, Serialization **Networking:** Introduction, Socket, Server socket, Client –Server Communication
3	UNIT 3
	Wrapper Classes: Introduction, Byte, Short, Integer, Long, Float, Double, Character, Boolean classes **Collection Framework**: Introduction, util Package interfaces, List, Set, Map, List interface & its classes, Set interface & its classes, Map interface & its classes **Inner Classes**: Introduction, Member inner class, Static inner class, Local inner class, Anonymous inner class **AWT:** Introduction, Components, Event-Delegation-Model, Listeners, Layouts, Individual components Label, Button, CheckBox, Radio Button, Choice, List, Menu, Text Field, Text Area

Reference Books

Core Java

Textbook(s):

1) Herbert Schildt, Java The Complete Reference, Ninth Edition, McGraw-Hill Education, 2014

Additional Reference(s):

1) E. Balagurusamy, Programming with Java, Tata McGraw-Hill Education India, 2014
2) Programming in JAVA, 2nd Ed, Sachin Malhotra & Saurabh Choudhary, Oxford Press
3) The Java Tutorials: http://docs.oracle.com/javase/tutorial/

Practical (1732UCSPR)

1. Accept integer values for a, b and c which are coefficients of quadratic equation. Find the solution of quadratic equation.
2. Accept two n x m matrices. Write a Java program to find addition of these matrices.
3. Accept n strings. Sort names in ascending order.
4. Create a package: Animals. In package animals create interface Animal with suitable behaviors. Implement the interface Animal in the same package animals.
5. Demonstrate Java inheritance using extends keyword.
6. Demonstrate method overloading and method overriding in Java.
7. Demonstrate creating your own exception in Java.
8. Using various swing components design Java application to accept a student's resume. (Design form)
9. Write a Java List example and demonstrate methods of Java List interface.
10. Design simple calculator GUI application using AWT components.

Course Code	Course	Hrs. of Instruction /Week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1733UCSOS	Core Subject Operating System	3	2 ½ Hours	25	75	100	2

Sr. No.	Modules / Units
1	UNIT 1
	Introduction and Operating-Systems Structures: Definition of Operating system, Operating System's role, Operating-System Operations, Functions of Operating System, Computing Environments **Operating-System Structures:** Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, Operating-System Structure **Processes**: Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication **Threads:** Overview, Multicore Programming, Multithreading Models
2	UNIT 2
	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 **CPU Scheduling:** Basic Concepts, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, SRTF, Priority, RR, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling), Thread Scheduling **Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock
3	UNIT 3
	Main Memory: Background, Logical address space, Physical address space, MMU, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table **Virtual Memory:** Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing **Mass-Storage Structure:** Overview, Disk Structure, Disk Scheduling, Disk Management **File-System Interface:** File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing **File-System Implementation:** File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management

Reference Books

Operating system

Textbook(s):

1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley,8th Edition

Additional Reference(s):

1. Achyut S. Godbole, Atul Kahate, Operating Systems, Tata McGraw Hill
2. Naresh Chauhan, Principles of Operating Systems, Oxford Press
3. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 4e Fourth Edition, Pearson Education, 2016

Practical (1733UCSPR)

Practical can be implemented either in JAVA or any other programming language.

1. Process Communication:

(i) Give solution to the producer–consumer problem using shared memory.

(ii) Give solution to the producer–consumer problem using message passing.

(iii) One form of communication in a Client–Server Systems environment is Remote method invocation (RMI). RMI is a Java feature similar to RPCs. RMI allows a thread to invoke a method on a remote object. Objects are considered remote if they reside in a different Java virtual machine (JVM). Demonstrate RMI program for adding/subtracting/multiplying/dividing two numbers.

2. Threads:

(i) The Java version of a multithreaded program that determines the summation of a non-negative integer. The Summation class implements the Runnable interface. Thread creation is performed by creating an object instance of the Thread class and passing the constructor a Runnable object.

(ii) Write a multithreaded Java program that outputs prime numbers. This program should work as follows: The user will run the program and will enter a number on the command line. The program will then create a separate thread that outputs all the prime numbers less than or equal to the number entered by the user.

(iii) The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5. 8, ... Formally, it can be expressed as: $fib_0 = 0$, $fib_1 = 1$, $fib_n = fib_{n-1} + fib_{n-2}$ Write a multithreaded program that generates the Fibonacci sequence using either the Java,

3. Synchronization:

(i) Give Java solution to Bounded buffer problem.

(ii) Give solution to the readers–writers problem using Java synchronization.

(iii) The Sleeping-Barber Problem: A barber shop consists of awaiting room with *n* chairs and a barber room with one barber chair. If there are no customers to be served, the barber goes to sleep. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy but chairs are available, then the customer sits in one of the free chairs. If the barber is asleep, the customer wakes up the barber. Write a program to coordinate the barber and the customers using Java synchronization.

4. Implement FCFS scheduling algorithm in Java.

5. Implement SJF (with no preemption) scheduling algorithm in Java

6. Implement RR scheduling algorithm in Java

7. Write a Java program that implements the banker's algorithm
8. Write a Java program that implements the FIFO page-replacement algorithm.
9. Write a Java program that implements the LRU page-replacement algorithm.

10. Design a File System in Java.

Course Code	**Course**	**Hrs. of Instruction /Week**	**Exam Duration (Hours)**	**Maximum Marks**			**Credits**
				CIE	**SEE**	**Total**	
1734UCSDB	**Core Subject** **Database Management System**	**3**	**$2^{1/2}$ Hours**	**25**	**75**	**100**	**2**

Sr. No.	**Modules / Units**
1	**UNIT 1**
	Stored Procedures: Types and benefits of stored procedures, creating stored procedures, executing stored procedures, altering stored procedures, viewing stored procedures. **Triggers:** Concept of triggers, Implementing triggers – creating triggers, Insert, delete, and update triggers, nested triggers, viewing, deleting and modifying triggers, and enforcing data integrity through triggers. **Sequences**: creating sequences, referencing, altering and dropping a sequence. **File Organization and Indexing:** Cluster, Primary and secondary indexing, Index data structure: hash and Tree based indexing, Comparison of file organization: cost model, Heap files, sorted files, clustered files. Creating, dropping and maintaining indexes.
2	**UNIT 2**
	Fundamentals of PL/SQL: Defining variables and constants, PL/SQL expressions and comparisons: Logical Operators, Boolean Expressions, CASE Expressions Handling, Null Values in Comparisons and Conditional Statements, PL/SQL Datatypes: Number Types, Character Types, Boolean Type, Datetime and Interval Types. **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
3	**UNIT 3**
	Transaction Management: ACID Properties, Serializability, Two-phase Commit Protocol, Concurrency Control, Lock Management, Lost Update Problem,

	Inconsistent Read Problem , Read-Write Locks, Deadlocks Handling, Two Phase Locking protocol. **DCL Statements:** Defining a transaction, Making Changes Permanent with COMMIT, Undoing Changes with ROLLBACK, Undoing Partial Changes with SAVEPOINT and ROLLBACK **Crash Recovery:** ARIES algorithm. The log based recovery, recovery related structures like transaction and dirty page table, Write-ahead log protocol, check points, recovery from a system crash, Redo and Undo phases.

Reference Books

Database Management System

Textbook(s):

1) Ramakrishnam, Gehrke, Database Management Systems, Bayross, McGraw-Hill,3rd Edition
2) Abraham Silberschatz, Henry F. Korth,S. Sudarshan , Database System Concepts, 6th Edition
3) Ivan Bayross, “SQL,PL/SQL -The Programming language of Oracle”, B.P.B. Publications

Additional Reference(s):

1) Ramez Elmasri & Shamkant B.Navathe, Fundamentals of Database Systems, Pearson Education
2) Robert Sheldon, Geoff Moes, Begning MySQL, Wrox Press.
3) Joel Murach, Murach’s MySQL, Murach

Practical (1734UCSPR)

Creating and working with Insert/Update/Delete Trigger using Before/After clause.
2. Writing PL/SQL Blocks with basic programming constructs by including following:
a. Sequential Statements b. unconstrained loop
3. Sequences:
a. Creating simple Sequences with clauses like START WITH, INCREMENT BY, MAXVALUE, MINVALUE, CYCLE | NOCYCLE, CACHE | NOCACHE, ORDER | NOORECER.
b. Creating and using Sequences for tables.
4. Writing PL/SQL Blocks with basic programming constructs by including following:
a. If...then...Else, IF...ELSIF...ELSE... END IF
b. Case statement
5. Writing PL/SQL Blocks with basic programming constructs for following Iterative Structure:
a. While-loop Statements
b. For-loop Statements.
6. Writing PL/SQL Blocks with basic programming constructs by including a GoTO to jump out of a loop and NULL as a statement inside IF
7. Writing Procedures in PL/SQL Block
a. Create an empty procedure, replace a procedure and call procedure
b. Create a stored procedure and call it
c. Define procedure to insert data
d. A forward declaration of procedure
8. Writing Functions in PL/SQL Block.
a. Define and call a function

b. Define and use function in select clause, c. Call function in dbms_output.put_line d. Recursive function e. Count Employee from a function and return value back f. Call function and store the return value to a variable 9. Writing a recursive Functions in PL/SQL Block 10. Study of transactions and locks

Course Code:	Course	Hrs. of Instruction/ week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1735UCSCG	Core Subject Combinatorics and Graph Theory	3	2 ½ hrs	25	75	100	2

Sr. No.	Modules / Units
1	**UNIT 1**
	Introduction to Combinatorics: Enumeration, Combinatorics and Graph Theory/ Number Theory/Geometry and Optimization, Sudoku Puzzles. Strings, Sets, and Binomial Coefficients: Strings- A First Look, Combinations, Combinatorial, The Ubiquitous Nature of Binomial Coefficients, The Binomial, Multinomial Coefficients. Induction: Introduction, The Positive Integers are Well Ordered, The Meaning of Statements, Binomial Coefficients Revisited, Solving Combinatorial Problems Recursively, Mathematical Induction, and Inductive Definitions Proofs by Induction. Strong Induction
2	**UNIT 2**
	Graph Theory: Basic Notation and Terminology, Multigraphs: Loops and Multiple Edges, Eulerian and Hamiltonian Graphs, Graph Coloring, Planar Counting, Labeled Trees, A Digression into Complexity Theory. Applying Probability to Combinatorics, Small Ramsey Numbers, Estimating Ramsey Numbers, Applying Probability to Ramsey Theory, Ramsey’s Theorem The Probabilistic Method
3	**UNIT 3**
	Network Flows: Basic Notation and Terminology, Flows and Cuts, Augmenting Paths, The Ford-Fulkerson Labeling Algorithm, A Concrete Example, Integer Solutions of Linear Programming Problems. Combinatorial Applications of Network Flows: Introduction, Matching in

	Bipartite Graphs, Chain partitioning, Pólya's Enumeration Theorem: Coloring the Vertices of a Square.

Reference Books

Combinatorics and Graph Theory

Textbook(s):

1) Applied Combinatorics, Mitchel T. Keller and William T. Trotter, 2016, http://www.rellek.net/appcomb.

Additional Reference(s):

1) Applied Combinatorics, sixth.edition, Alan Tucker, Wiley; (2016)
2) Graph Theory and Combinatorics, Ralph P. Grimaldi, Pearson Education; Fifth edition (2012)
3) Combinatorics and Graph Theory, John Harris, Jeffry L. Hirst, Springer(2010).
4) Graph Theory: Modeling, Applications and Algorithms, Agnarsson, Pearson Education India (2008).

Practical (1735UCSPR)

1. Solving problems on strings, sets and binomial coefficients.
2. Solving problems using induction.
3. Solving problems on Eulerian and Hamiltonian graphs.
4. Solving problems on Chromatic number and coloring
5. Solving problems using Kruskal's Algorithm
6. Solving problems using Prim's Algorithm
7. Solving problems using Dijkstra's Algorithm
8. Solving problems of finding augmenting paths in network flows.
9. Solving problems on network flows using Ford-Fulkerson Labeling Algorithm
10. Solving problems on posets and their associated networks.

Course Code:	Course	Hrs. of Instruction/ week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1736UCSPC	Core Subject: Physical Computing and IoT	3	2 ½ hrs	25	75	100	2

Sr. No.	Modules / Units
1	**UNIT 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. **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.
2	**UNIT 2**
	Programming Raspberry Pi **Raspberry Pi and Linux:** About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands **Programing interfaces:** Introduction to Node.js, Python. **Raspberry Pi Interfaces:** UART, GPIO, I2C, SPI **Useful Implementations:** Cross Compilation, Pulse Width Modulation, SPI for Camera.
3	**UNIT 3**
	Introduction to IoT: What is IoT? IoT examples, Simple IoT LED Program. **IoT and Protocols** **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.

Reference Books

Combinatorics and Graph Theory

Textbook(s):

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 (1736UCSPR)
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

Course Code:	Course	Hrs. of Instruction/ week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1737UCSWP	Skill Enhancement: Web programming	3	2 ½ hrs	25	75	100	2

Sr. No.	Modules / Units
1	UNIT 1
	HTML5: Fundamental Elements of HTML, Formatting Text in HTML, Organizing Text in HTML, Links and URLs in HTML, Tables in HTML, Images on a Web Page, Image Formats, Image Maps, Colors, FORMs in HTML, Interactive Elements, Working with Multimedia - Audio and Video File Formats, HTML elements for inserting Audio / Video on a web page **CSS:** Understanding the Syntax of CSS, CSS Selectors, Inserting CSS in an HTML Document, CSS properties to work with background of a Page, CSS properties to work with Fonts and Text Styles, CSS properties for positioning an element
2	UNIT 2
	JavaScript: Using JavaScript in an HTML Document, Programming Fundamentals of JavaScript – Variables, Operators, Control Flow Statements, Popup Boxes, Functions – Defining and Invoking a Function, Defining Function arguments, Defining a Return Statement, Calling Functions with Timer, JavaScript Objects - String, RegExp, Math, Date, Browser Objects - Window,

	Navigator, History, Location, Document, Cookies, Document Object Model, Form Validation using JavaScript **XML:** Comparing XML with HTML, Advantages and Disadvantages of XML, Structure of an XML Document, XML Entity References, DTD, XSLT: XSLT Elements and Attributes - xsl:template, xsl:apply-templates, xsl:import, xsl:call-template, xsl:include, xsl:element, xsl:attribute, e xsl:attribute-set, xsl:value-of
3	**UNIT 3**
	AJAX: AJAX Web Application Model, How AJAX Works, XMLHttpRequest Object – Properties and Methods, Handling asynchronous requests using AJAX **PHP:** Variables and Operators, Program Flow, Arrays, Working with Files and Directories, Working with Databases, Working with Cookies, Sessions and Headers **Introduction to jQuery:** Fundamentals, Selectors, methods to access HTML attributes, methods for traversing, manipulators, events, effects

Reference Books

Web Programming

Text Book(s):

1) HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press
2) Web Programming and Interactive Technologies, scriptDemics, StarEdu Solutions India.
3) PHP: A Beginners Guide, Vikram Vaswani, TMH

Additional Reference(s):

1) HTML, XHTML, and CSS Bible Fifth Edition, Steven M. Schafer, WILEY
2) Learn to Master HTML 5, scriptDemics, StarEdu Solutions Pvt Ltd.
3) Learning PHP, MySQL, JavaScript, CSS & HTML5, Robin Nixon, O'Reilly
4) PHP, MySQL, JavaScript & HTML5 All-in-one for Dummies, Steve Suehring, Janet Valade Wiley

Practical (1737UCSPR)

1. Design a webpage that makes use of
a. Document Structure Tags b. Various Text Formatting Tags
c. List Tags d. Image and Image Maps
2. Design a webpage that makes use of
a. Table tags b. Form Tags (forms with various form elements)

c. Navigation across multiple pages d. Embedded Multimedia elements
3. Design a webpage that make use of Cascading Style Sheets with
a. CSS properties to change the background of a Page
b. CSS properties to change Fonts and Text Styles
c. CSS properties for positioning an element

4. Write JavaScript code for

a. Performing various mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number

b. Validating the various Form Elements

5. Write JavaScript code for

a. Demonstrating different JavaScript Objects such as String, RegExp, Math, Date

b. Demonstrating different JavaScript Objects such as Window, Navigator, History, Location, Document,

c. Storing and Retrieving Cookies

6. Create a XML file with Internal / External DTD and display it using

a. CSS b. XSL

7. Design a webpage to handle asynchronous requests using AJAX on

a. Mouseover b. button click

8. Write PHP scripts for

a. Retrieving data from HTML forms

b. Performing certain mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number

c. Working with Arrays

d. Working with Files (Reading / Writing)

9. Write PHP scripts for

a. Working with Databases (Storing Records / Reprieving Records and Display them)

b. Storing and Retrieving Cookies

c. Storing and Retrieving Sessions

10. Design a webpage with some jQuery animation effects.

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)