Learning Outcome based Curriculum Framework (LOCF)

For

Choice Based Credit System (CBCS)

Syllabus

B.Sc. (Program) in Computer Science

w.e.f. Academic Session 2020-21

Kazi Nazrul University

Asansol, Paschim Bardhaman
West Bengal 713340

PREAMBLE ACCORDING TO UGC LOCF to be included

Semester- I

Course Name: PROGRAMMING METHODOLOGIES

Course Code: BSCPCOSC101

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-1(1) or CC-2(1) or CC-3(1)**		L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Learn to develop simple algorithms and flow charts to solve a problem.*
2. *Develop problem solving skills coupled with top down design principles.*
3. *Learn about the strategies of writing efficient and well-structured computer algorithms/programs.*
4. *Develop the skills for formulating iterative solutions to a problem.*
5. *Learn array processing algorithms coupled with iterative methods.*
6. *Learn text and string processing efficient algorithms.*
7. *Learn searching techniques and use of pointers.*

Course Content:

Theory

UNIT I. Introduction to Programming, Program Concept, Characteristics of Programming, Stages in Program Development, Algorithms, Notations, Design, Flowcharts, Types of Programming Methodologies, Introduction to C/C++ Programming - Basic Program Structure In C/C++, Variables and Assignments, Input and Output, Selection and Repetition Statements.

UNIT II. Top-Down Design, Predefined Functions, Programmer -defined Function, Local Variable, Functions with Default Arguments

UNIT III. Introduction to Arrays, Declaration and Referring Arrays, Arrays in Memory, Initializing Arrays. Arrays in Functions, Multi-Dimensional Arrays.

UNIT IV. Pointers - Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Call-By-Value and Call-By-Reference Parameters.

UNIT V. Structures - Member Accessing, Pointers to Structures, Structures and Functions, Arrays of Structures, Unions.

UNIT VI. Strings - Declaration and Initialization, Reading and Writing Strings, Arrays of Strings, String and Function, Strings and Structure, Standard String Library Functions.

UNIT VII. Searching Algorithms - Linear Search and Binary Search.

Practical

UNIT I. Given the problem statement, students are required to formulate problem, develop flowchart/algorithm, write code, execute and test it. Students should be given assignments on following :

a) To learn elementary techniques involving arithmetic operators and mathematical expressions, appropriate use of selection (if, switch, conditional operators) and control structures

b) Learn how to use functions and parameter passing in functions.

UNIT II. Students should be given assignments on following:

a) Write Programs to learn the use of strings and string handling operations.

b) Problems which can effectively demonstrate use of Arrays. Structures and Union.

c) Write programs using pointers.

d) Write programs to implement search algorithms.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks) – one from each unit, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from each unit, Viva-voce (10 marks)

References/ Suggested Readings:

1. Problem Solving and Program Design in C, J. R. Hanly and E. B. Koffman, Pearson, 2015.
2. Programming and problem solving with C++: brief edition, N. Dale and C. Weems, Jones & Bartlett Learning, 2010.
3. C Programming, Karnighan,&Ritchie, PHI
4. Programming through C, Richard Johnsonbaugh and Martin Kalin, Pearson Education
5. Programming in C, B.S. Gottfried, Sahaum Series.
6. Programming in ANSI C, E. Balaguruswami, TMH

Semester- II

Course Name: DATA STRUCTURE

Course Code: BSCPCOSC201

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-1(2) or CC-2(2) or CC-3(2)**		L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *To be familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles*
2. *To have knowledge of complexity of basic operations like insert, delete, search on these data structures.*
3. *Ability to choose a data structure to suitably model any data used in computer applications.*
4. *Design programs using various data structures Binary and general search trees, heaps etc.*
5. *Ability to assess efficiency tradeoffs among different data structure implementations.*
6. *Implement and know the applications of algorithms for sorting, searching etc.*

Course Content:

Theory

UNIT I. Basic concepts- Algorithm Specification-Introduction, Recursive algorithms, Data Abstraction, Performance analysis, Linear and Non Linear data structures, Singly Linked Lists-Operations, Concatenating, Circularly linked lists-Operations for Circularly linked lists, Doubly Linked Lists- Operations. Representation of single, two dimensional arrays.

UNIT II. Stack- Definition and Operations, Array and Linked Implementations, Applications - Valid Expression Checking (Parenthesis matching), Reversal of string, Infix to Postfix Conversion, Postfix Expression Evaluation, Recursion Implementation.

UNIT III. Queue - Definition and Operations, Array and Linked Implementations, Applications, Circular Queues - Insertion and Deletion Operations, Dequeue (Double Ended Queue) - Introduction.

UNIT IV. Sorting Methods – Bubble, Insertion, Selection, Using Divide-Conquer Approach (Quick and Merge sort), Comparison of Sorting Methods, Searching Methods – Linear and Binary.

UNIT V. Trees, Representation of Trees, Binary tree, Properties of Binary Trees, Binary Tree Representations- Array and Linked Representations, Binary Tree Traversals, Threaded Binary Trees, Binary Search tree - Creation, Insertion, Deletion and Search, Heap-Definition, Min heap, Max heap, Insertion and Deletion.

Practical

Students are required to write and practically execute programs to solve problem using various data structures. The teacher can suitably device problems which help students experiment using the suitable data structures and operations. Some of the problems are indicated below.

1. Write program that uses functions to perform the following:
   a) Creation of list of elements where the size of the list, elements to be inserted and deleted are dynamically given as input.
   b) Implement the operations, insertion, deletion at a given position in the list and search for an element in the list
   c) To display the elements in forward / reverse order.

2. Write a program to implement stack data structure and basic operations on it (Insertion, deletion). Write a program that demonstrates the application of stack operations (Eg: infix expression to postfix conversion, postfix evaluation).

3. Write a program to implement queue data structure and basic operations on it (Insertion, deletion, find length) and code at least one application using queues.

4. Write program that implements linear and binary search methods of searching for an elements in a list.

5. Write and trace programs to understand the various phases of sorting elements using the methods a) Bubble sort b) Insertion Sort c) Quicksort etc.

6. Write a program to create a Binary search tree and insert and delete from the tree. Write recursive and non-recursive routines to traverse a binary tree in preorder, inorder and postorder.

7. Write programs for recursion (Eg. Fibonacci numbers).

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks), Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks), Viva-voce (10 marks)

References/ Suggested Readings:

1. Fundamentals of Data structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson-Freed, Universities Press.
2. Data structures and Algorithm Analysis in C, 2nd edition, M. A. Weiss, Pearson.
3. Lipschutz: Schaum's outline series Data structures Tata McGraw-Hill
4. Data Structure through C in Depth. S.K. Srivastava and Deepali Srivastava, B.P.B Publication.

Semester- III

Course Name: Basics of Operating Systems

Course Code: BSCPCOSC301

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-1(3) or CC-2(3) or CC-3(3)**		L-T-P: **5 - 1 - 0**	
	Full Marks: **50**	CA Marks		ESE Marks	
Credit: **6**		Practical	Theoretical	Practical	Theoretical
		-----	**10**	-----	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Describe the important computer system resources and the role of operating system in their management policies and algorithms.*
2. *To understand various functions, structures and history of operating systems and should be able to specify objectives of modern operating systems and describe how operating systems have evolved over time.*
3. *Understanding of design issues associated with operating systems.*
4. *Understand various process management concepts including scheduling, synchronization, and deadlocks.*
5. *To have a basic knowledge about multithreading.*
6. *To understand concepts of memory management including virtual memory.*
7. *To understand issues related to file system interface and implementation, disk management.*
8. *To have sound knowledge of various types of operating systems including Unix and Android.*
9. *Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve.*

Course Content:

Theory

UNIT I. (Introduction to Operating System) What is Operating System? History and Evolution of OS, Basic OS functions, Resource Abstraction, Types of Operating Systems– Multiprogramming Systems, Batch Systems, Time Sharing Systems; Operating Systems for Personal Computers, Workstations and Hand-held Devices, Process Control & Real time Systems.

UNIT II. (Operating System Organization and Process Characterization) Processor and User Modes, Kernels, System Calls and System Programs, System View of the Process and

Resources, Process Abstraction, Process Hierarchy, Threads, Threading Issues, Thread Libraries; Process Scheduling, Non-Pre-emptive and Preemptive Scheduling Algorithms.

UNIT III. Process Management (Deadlock) Deadlock, Deadlock Characterization, Necessary and Sufficient Conditions for Deadlock, Deadlock Handling Approaches: Deadlock Prevention, Deadlock Avoidance and Deadlock Detection and Recovery.

UNIT IV. (Inter Process Communication and Synchronization) Concurrent and Dependent Processes, Critical Section, Semaphores, Methods for Inter-process Communication; Process Synchronization, Classical Process Synchronization Problems: Producer-Consumer, Reader-Writer.

UNIT V. (Memory Management) Physical and Virtual Address Space; Memory Allocation Strategies– Fixed and -Variable Partitions, Paging, Segmentation, Virtual Memory; Page Replacement Algorithms.

UNIT VI. (File and I/O Management, Disk Scheduling) Directory Structure, File Operations, File Allocation Methods, Device Management, Pipes, Buffer, Shared Memory, Disk Scheduling (FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK).

References/ Suggested Readings:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.
2. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
3. W. Stallings, Operating Systems, Internals & Design Principles 2008 5th Edition, Prentice Hall of India. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

Course Name: Core Java

Course Code: BSCPCOSSE301

Course Type: SE (Practical)	Course Details: SEC-1		L-T-P: 0 - 0 - 8	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**-----**	**20**	**----**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Knowledge of the structure and model of the Java programming language,*

2. *Use the Java programming language for various programming technologies*

3. *Develop software in the Java programming language,*

4. *Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements.*

Course Content:

Practical

UNIT I. Introduction: Java Essentials, Its characteristics, Execution and Compilation, Data types, Variables, Control Statements, Standard Input/ Output.

UNIT II. Constructors, Object Oriented Concepts: Encapsulation, Abstraction, Inheritance, Polymorphisms, JAVA Packages.

UNIT III. Exception Handling, Wrapper Classes, Autoboxing.

Students are required to implement object-oriented paradigm using JAVA. Below is the list of some of the experiments.

Part A

1. Program on strings: Check the equality of two strings, Reverse a string.

2. Program using loops: to find the sum of digits of a given number, display a multiplication table, display all prime numbers between 1 to 1000.

3. Program to demonstrate all math class functions.

Part B

4. Program on files: to copy a file to another file using Java to package classes.

5. Program to demonstrate method over-riding and overloading

6. Programs on inheritances.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two experiments (10 marks) – one from Part A and another from Part B, Viva-voce (5 marks)

ESE Evaluation: Two experiments (10 marks) – one from Part A and another from Part B, Viva-voce (10 marks)

References/ Suggested Readings:

1. E. Balaguruswamy, Programming with Java, 4th Edition, McGraw Hill.2009.
2. John R. Hubbard,"Programming with JAVA, Schaum's Series, 2nd Edition, 2004.
3. Herbert Schildt, The Complete Reference Java 2, TMH.

Semester- IV

Course Name: Basics of Database Management System

Course Code: BSCPCOSC401

Course Type: **Core (Theoretical & Practical)**	Course Details: **CC-1(4) or CC-2(4) or CC-3(4)**		L-T-P: **4 - 0 - 4**	
Credit: **6**	Full Marks: **100**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	**20**	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. Gain knowledge of database systems and database management systems software

2.   Ability to model data in applications using conceptual modelling tools such as ER Diagrams and design data base schemas based on the model.

3. Formulate, using SQL, solutions to a broad range of query and data update problems.

4. Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.

5. Be acquainted with the basics of transaction processing and concurrency control.

6. Familiarity with database storage structures and access techniques.

Course Content:

Theory

UNIT I. Basic Database Concepts, Terminology, and Architecture; Types of Database Management Systems. Differences between Relational and other Database Models. Data Modelling: Relations, Schemas, Constraints, Queries, and Updates; Conceptual vs. Physical Modeling; Entity Types, attributes, ER Diagrams.

UNIT II. SQL Data Definition: Specifying Tables, Data Types, Constraints; Simple SELECT, INSERT, UPDATE, DELETE Statements; Complex SELECT Queries, including Joins and Nested Queries; Actions and Triggers; Views.

UNIT III. Relational Algebra: Definition of Algebra; Relations as Sets; Operations: SELECT, PROJECT, JOIN, etc. Normalization Theory and Functional Dependencies, 2NF, 3NF, BCNF;

UNIT IV. Indexing: Files, Blocks, and Records, Hashing; RAID; Replication; Single-Level and Multi-Level Indexes; B-Trees and B+-Trees. Basics of Transactions, Concurrency and Recovery.

Practical

Students are required to practice the concepts learnt in the theory by designing and querying a database for a chosen organization (Like Library, Transport etc). The teacher may devise appropriate weekly lab assignments to help students practice the designing, querying a database in the context of example database. Some indicative list of experiments is given below.

Experiment 1: E-R Model Analyze the organization and identify the entities, attributes and relationships in it. . Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Experiment 2: Concept design with E-R Model Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any).

Experiment 3: Relational Model Represent all the entities (Strong, Weak) in tabular fashion. Represent relation ships in a tabular fashion.

Experiment 4: Normalization Apply the First, Second and Third Normalization levels on the database designed for the organization

Experiment 5: Practicing DDL commands. Creating databases, How to create tables, altering the database, dropping tables and databases. Try truncate, rename commands etc.

Experiment 6: Practicing DML commands on the Database created for the example organization DML commands are used to for managing data within schema objects. Some examples: ● SELECT - retrieve data from the a database ● INSERT - insert data into a table ● UPDATE - updates existing data within a table ● DELETE - deletes all records or few records from a table.

Experiment 7: Practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

Internal (CA) Evaluation: Practical Note Book (15 marks), Experiment (10 marks) – Experiment 1 to Experiment 4, Viva-voce (5 marks)

ESE Evaluation: Experiment (10 marks) – Experiment 5 to Experiment 7, Viva-voce (10 marks)

References/ Suggested Readings:

1. Elmasri's and Navathe's Fundamentals of Database Systems. Addison-Wesley
2. Silberschatz, Henry. F. Korth, S. Sudarshan, Data base System Concepts, McGraw Hill Education
3. A. Majumdar and P. Bhattacharyya, Database Management Systems, McGraw Hill Education.

Course Name: Web Programming with PHP

Course Code: BSCPCOSSE401

Course Type: SE (Practical)	Course Details: **SEC-2**		L-T-P: **0 - 0 - 8**	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**-----**	**20**	**----**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):
1. *To understand basics of the Internet and World Wide Web*
2. *To acquire knowledge and skills for creation of web site considering both client and server-side programming*
3. *To learn basic skill to develop responsive web applications*
4. *To understand different web extensions and web services standards*
5. *To understand basic concepts of Search Engine Basics.*
6. *To learn Web Service Essentials.*

Course Content:

Practical

Students are directed to do a minor project based on the contents of the course below (UNIT I to UNIT V) for internal and ESE evaluation.

UNIT I. (Introduction to World Wide Web) -Internet Standards, Introduction to WWW and WWW Architecture, Internet Protocols, Overview of HTTP, HTTP request – response, Generations of dynamic web pages.

UNIT II. (User Interface Design) Introduction to HTML and HTML5, TML Tags, Formatting and Fonts, Commenting Code, Anchors, Backgrounds, Images, Hyperlinks, Lists, Tables, Frames, HTML Forms. The need for CSS, Introduction to CSS, Basic syntax and structure, Inline Styles, Embedding Style Sheets, Linking External Style, Backgrounds, Manipulating Text, Margins and Padding, Positioning using CSS.

UNIT III. (Java Script) Introduction to Java Script, Java Script variables, operators, decision control statements, Looping and arrays. Functions and events, Popup boxes-alert, prompt, conform box, Built-in objects, writing JavaScript form validation.

UNIT IV. (PHP) Introduction to PHP**,** Server side scripting, Role of web server software, PHP comments, variables, echo and print, PHP operators, data types , Branching statements, Loops, Arrays, PHP functions, PHP form, Passing information between pages, $_GET, $_POST, $_REQUEST., String functions, include and require, session and cookie management, Error handling in PHP, Object Oriented Programming using PHP.

UNIT V. (PHP with MYSQL) Introduction to MySQL, datatypes, SQL commands-CREATE, UPDATE, INSERT, DELETE, SELECT, PHP functions for MySQL connectivity and operation- mysql_connect, mysql_select_db, mysql_query, Updation and deletion of data using PHP, Displaying data from MySQL in webpage.

Internal (CA) Evaluation: Minor Project Report (15 marks), Demonstration of the minor project (10 marks), Viva-voce (5 marks).

ESE Evaluation: Presentation of the minor project (10 marks), Viva-voce (10 marks).

References/ Suggested Readings:

1. Ivan Bayross ,"HTML, DHTML, JavaScript, Pearl & CGI", Fourth Revised Edition, BPB Publication.
2. Rasmus Lerdorf and Kevin Tatore ,"Programming PHP", Shroff Publishers & Distributors Pvt.Ltd.
3. Dave W Mercer, Allan Kent, Steven D Nowicki, David Mercer, Dan Squier, Wankyu Choi, "Beginning PHP", Wiley Publishing, In.
4. Robin Nixon, Learning Php, MySQL & JavaScript: A Step-By-Step Guide To Creating Dynamic Websites, Sixth Edition, O'REILLY.

Semester- V

Course Name: Basics of Software Engineering

Course Code: BSCPCOSDSE501

Course Type: **DSE** (Theoretical)	Course Details: **DSEC-1(1) or DSEC-2(1) or DSEC-3(1)**		L-T-P: **5 - 1 - 0**	
		CA Marks		ESE Marks
Credit: **6**	Full Marks: **50**	Practical	Theoretical	Practical	Theoretical
		….	**10**	….	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Basic knowledge and understanding of the analysis and design of complex systems.*
2. *Ability to apply software engineering principles and techniques.*
3. *To produce efficient, reliable, robust and cost-effective software solutions.*
4. *Ability to work as an effective member or leader of software engineering teams.*
5. *To manage time, processes and resources effectively by prioritizing competing demands to achieve personal and team goals Identify and analyzes the common threats in each domain.*

Course Content:

Theory

UNIT I. Software Development Approaches: Introduction; Evolving Role of Software; Software Characteristics; Software Applications. Software Design Processes: Introduction; What is Meant by Software Engineering? Definitions of Software Engineering; The Serial or Linear Sequential Development Model; Iterative Development Model; The incremental Development Model

UNIT II. Software Design Principles: Introduction, System Models: Data-flow Models, Semantic Data Models, Object Models, Inheritance Models, Object Aggregation, Service Usage Models, Data Dictionaries; Software Design: The Design Process, Design Methods, Design description, Design Strategies, Design Quality.

UNIT III. Object Oriented Design: Introduction; Object Oriented Design: Objects, Object Classes & Inheritance, Inheritance, Object Identification, An Object -Oriented Design Example, Object Aggregation; Service Usage; Object Interface Design: Design Evolution, Function Oriented Design, Data–Flow Design; Structural Decomposition: Detailed Design.

UNIT IV. Software Reliability: Introduction; Software Reliability Metrics; Programming for Reliability: Fault Avoidance, Fault Tolerance, Software Reuse.

UNIT V. Software Testing Techniques: Introduction; Software Testing Fundamental; Testing Principles; White Box Testing; Control Structure Testing; Black Box Testing; Boundary Value Analysis; Testing GUIs; Testing Documentation and Help Facilities; Software Testing Strategies: Introduction; Organizing for Software Testing; Software Testing Strategy, Unit Testing: Unit Test Considerations, Top-Down Integration, Bottom-Up Integration.

References/ Suggested Readings:

1. R. G. Pressman, Software Engineering, TMH.
2. Rajib Mall, Fundamentals of Software Engineering, PHI Learning Pvt. Ltd.
3. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publications. Pfleeger, Shari Lawrence, Software Engineering Theory and Practice, Second edition. Prentice- Hall.

Course Name: Basics of Computer Networks

Course Code: BSCPCOSDSE502

Course Type: **DSE** (Theoretical)	Course Details: **DSEC-1(1) or DSEC-2(1) or DSEC-3(1)**		L-T-P: **5 - 1 - 0**	
Credit: **6**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		….	**10**	….	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Understand the structure of Data Communications System and its components. Be familiarize with different network terminologies.*
2. *Familiarize with contemporary issues in network technologies.*
3. *Know the layered model approach explained in OSI and TCP/IP network models*
4. *Identify different types of network devices and their functions within a network.*
5. *Learn basic routing mechanisms, IP addressing scheme and internetworking concepts.*
6. *Familiarize with IP and TCP Internet protocols.*
7. *To understand major concepts involved in design of WAN, LAN and wireless networks.*
8. *Learn basics of network configuration and maintenance.*

Course Content:

Theory

UNIT I. Introduction to Computer Networks and Networking Elements: Network Definition, Network Topologies, Network Classifications, Network Protocol, Layered Network

Architecture, Overview of OSI Reference Model, Overview of TCP/IP Protocol Suite, Hub, Switch (Managed and Unmanaged), Routers.

UNIT II. Data Communication Fundamentals and Techniques: Analog and Digital Signal, Data-Rate Limits, Digital to Digital Line Encoding Schemes, Pulse Code Modulation, Parallel and Serial Transmission, Digital to Analog Modulation - Multiplexing Techniques- FDM, TDM, Transmission Media.

UNIT III. Networks Switching Techniques and Access Mechanisms: Circuit Switching, Packet Switching- Connectionless Datagram Switching, Connection Oriented Virtual Circuit Switching; Dial-Up Modems, Digital Subscriber Line, Cable TV for Data Transfer.

UNIT IV. Data Link Layer Functions and Protocol: Error Detection and Error Correction Techniques, Data-Link Control- Framing and Flow Control, Error Recovery Protocols-Stop and Wait ARQ, Go-Back-N ARQ, Point to Point Protocol on Internet.

UNIT V. Multiple Access Protocol and Network Layer: CSMA/CD Protocols, Ethernet LANS; Connecting LAN and Back-Bone Networks- Repeaters, Hubs, Switches, Bridges, Router and Gateways, Networks Layer Functions and Protocols, Routing, Routing Algorithms, Network Layer Protocol of Internet - IP Protocol, Internet Control Protocols.

UNIT VI. Transport Layer and Application Layer Functions and Protocols: Transport Services- Error and Flow Control, Connection Establishment and Release- Three Way Handshake, Overview of Application Layer Protocol, Overview of DNS Protocol; Overview of WWW & HTTP Protocol.

References/ Suggested Readings:

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM Publishing Company Ltd 2007.
2. A. S. Tanenbaum: Computer Networks, Fifth edition, PHI Pvt. Ltd 2011
3. William Stallings:Data and Computer Communications, Eight Edition, Pearson.

.

Course Name: MATLAB Programming

Course Code: BSCPCOSSE501

Course Type: SE (Practical)	Course Details: **SEC-3**		L-T-P: **0 - 0 - 8**	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	-----	**20**	----

Course Learning Outcomes:
(After the completion of course, the students will have ability to):

1. *Understand the fundaments of procedural and functional programming;*
2. *Understand Matlab data types and structures;*
3. *Be able to set up simple real-life numerical problems such that they can be solved and visualized using basic codes in Matlab;*
4. *Be ready to use advanced coding in Matlab in their subsequent studies*

Course Content:

Practical

UNIT I. Introduction to MATLAB Programming- Basics of MATLAB programming, Array operations in MATLAB, Loops and execution control, Working with files: Scripts and Functions, Plotting and program output.

UNIT II. Approximations and Errors- Defining errors and precision in numerical methods, Truncation and round-off errors, Error propagation, Global and local truncation errors.

UNIT III. Linear Equations- Linear algebra in MATLAB, Gauss Elimination, LU decomposition and partial pivoting, Iterative methods: Gauss Siedel Method.

UNIT IV. Regression and Interpolation- Introduction, Linear least squares regression(including lsqcurvefit function), Functional and nonlinear regression (including lsqnonlin function), Interpolation in MATLAB using spline and pchip.

UNIT V. Nonlinear Equations- Nonlinear equations in single variable, MATLAB function fzero in single variable, Fixed-point iteration in single variable, NewtonRaphson in single variable, MATLAB function fsolve in single and multiple variables, Newton-Raphson in multiple variables.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two Experiments (10 marks), Viva-voce (5 marks)

ESE Evaluation: Two Experiments (10 marks), Viva-voce (10 marks).

References/ Suggested Readings:

1. Fausett L.V, Applied NumericalAnalysis Using MATLAB, Pearson Education.
2. Brian Hahn; Daniel T. Valentine, Essential MATLAB for Engineers and Scientists,, Academic Press.

Course Name: Programming in Python

Course Code: BSCPCOSSE502

Course Type: SE (Practical)	Course Details: SEC-3		L-T-P: 0 - 0 - 8	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	-----	**20**	----

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Develop and Execute simple Python programs.*
2. *Structure a Python program into functions.*
3. *Using Python lists, tuples to represent compound data*
4. *Develop Python Programs for file processing*

Course Content:

Practical

UNIT I. Introduction to Python, Python, Features of Python, Execution of a Python, Program, Writing Our First Python Program, Data types in Python. Python Interpreter and Interactive Mode; Values and Types: int, float, boolean, string, and list; Variables, Expressions, Statements, Tuple Assignment, Precedence of Operators, Comments; Modules and Functions, Function Definition and use, Flow of Execution, Parameters and Arguments

UNIT II. Operators in Python, Input and Output, Control Statements. Boolean Values and operators, Conditional (if), Alternative (if-else), Chained Conditional (if-elif-else); Iteration: state, while, for, break, continue, pass; Fruitful Functions: Return Values, Parameters, Local and Global Scope, Function Composition, Recursion

UNIT III. Arrays in Python, Strings and Characters. Strings: String Slices, Immutability, String Functions and Methods, String Module; Lists as Arrays. Illustrative Programs: Square Root, gcd, Exponentiation, Sum an Array of Numbers, Linear Search, Binary Search.

UNIT IV. Functions, Lists and Tuples. List Operations, List Slices, List Methods, List Loop, Mutability, Aliasing, Cloning Lists, List Parameters; Tuples: Tuple Assignment, Tuple as Return Value; Dictionaries: Operations and Methods.

UNIT V. Files and Exception: Text Files, Reading and Writing Files, Format Operator; Command Line Arguments, Errors and Exceptions, Handling Exceptions, Modules, Packages; Illustrative Programs: Word Count, Copy File.

The students are required to verify their ability to use core programming basics and program design with functions using Python programming language. The teacher shall programs to strengthen the practical expertise of the students. The following is an indicative list of programs that can be practised.

1. Write a program to demonstrate different number data types in Python.

2. Write a program to perform different Arithmetic Operations on numbers in Python.
3. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
4. Write a python script to print the current date in the following format "Sat Oct 11 02:26:23 IST 2020"
5. Write a program to create, append, and remove lists in python.
6. Write a program to demonstrate working with tuples in python.
7. Write a program to demonstrate working with dictionaries in python.
8. Write a python program to find largest of three numbers.
9. Write a Python program to construct the different pattern, using a nested for loop,
   Like
   ```
      *
      * *
      * * *
      * *
       *
   ```
10. Write a Python script that prints prime numbers less than 20.
11. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
12. Write a python program to define a module and import a specific function in that module to another program.
13. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
14. Write a Python class to convert an integer to a roman numeral.
15. Write a Python class to reverse a string word by word.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two Experiments (10 marks), Viva-voce (5 marks)

ESE Evaluation: Two Experiments (10 marks), Viva-voce (10 marks).

References/ Suggested Readings:

1. Allen Downey, Think Python, Green Tea Press.
2. Wesley J. Chun, Core Python Programming, Pearson Education.
3. Mark Lutz, Learning Python, Oreilly Publication.
4. Kenneth A. Lambert, Fundamentals of Python: First Programs, Course Technology Inc.

Semester- VI

Course Name: Basics of Artificial Intelligence

Course Code: BSCPCOSDSE601

Course Type: **DSE** (Theoretical & Practical)	Course Details: **DSEC-1(2) or DSEC-2(2) or DSEC-3(2)**		L-T-P: 4 **- 0 - 4**	
Credit: **6**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		30	**10**	20	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Explain what constitutes "Artificial" Intelligence and how to identify systems with Artificial Intelligence.*
2. *Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.*
3. *Formalize a given problem in the language/framework of different AI methods (e.g., as a search problem, as a constraint satisfaction problem, as a planning problem, etc).*
4. *Implement basic AI algorithms (e.g., standard search or constraint propagation algorithms).*
5. *Design and perform an empirical evaluation of different algorithms on a problem formalization, and state the conclusions that the evaluation supports.*
6. *Explain the limitations of current Artificial Intelligence techniques.*

Course Content:

Theory

UNIT I. Introduction to Artificial Intelligence: Definition of AI; Turing Test; Brief History of AI. Problem Solving and Search: Problem Formulation; Search Space; States vs. Nodes; Tree Search: Breadth-First, Uniform Cost, Depth-First, Depth-Limited, Iterative Deepening; Graph Search.

UNIT II. Informed Search: Greedy Search; A* Search; Heuristic Function; Admissibility and Consistency; Deriving Heuristics via Problem Relaxation. Local Search: Hill-Climbing; Simulated Annealing; Genetic Algorithms; Local Search in Continuous Spaces.

UNIT III. Playing Games: Game Tree; Utility Function; Optimal Strategies; Minimax Algorithm; Alpha-Beta Pruning; Games with an Element of Chance. Beyond Classical Search: Searching with Nondeterministic Actions; Searching with Partial Observations; Online Search Agents; Dealing with Unknown Environments.

UNIT IV. Knowledge Representation and Reasoning: Ontologies, Foundations of Knowledge Representation and Reasoning, Representing and Reasoning about Objects, Relations, Events, Actions, Time, and Space; Predicate Logic, Situation Calculus,

Description Logics, Reasoning with Defaults, Reasoning about Knowledge, Sample Applications.

UNIT V. Representing and Reasoning with Uncertain Knowledge: Probability, Connection to Logic, Independence, Bayes Rule, Bayesian Networks, Probabilistic Inference, and Sample Applications.

UNIT VI. Planning: The STRIPS Language; Forward Planning; Backward Planning; Planning Heuristics; Partial-Order Planning; Planning using Propositional Logic; Planning vs. Scheduling.

UNIT VII. Constraint Satisfaction Problems (CSPs): Basic Definitions; Finite vs. Infinite vs. Continuous Domains; Constraint Graphs; Relationship With Propositional Satisfiability, Conjunctive Queries, Linear Integer Programming, and Diophantine Equations; NP49 Completeness of CSP.

Practical

The students are expected to explore the foundational skills on AI techniques acquired in theory in solving problems using LISP.

UNIT I. Introduction: The need for symbolic computation. Why LISP is a good language for symbolic computation.

UNIT II. LISP Introduction: Atoms, lists, S-expressions, functions, lambdas, predicates, conditionals, recursion, iteration, printing, reading, properties, a-lists.

UNIT III. Search: Means-ends analysis, defining operators, blocks-world planning, Sussman Anomaly, interacting goals.

UNIT IV. Pattern Matching: Pattern matching, rule-based translation, a simplistic natural-language dialog system.

UNIT V. Search Tools: A general search program. Heuristic search, best-first search, beam-search, hill-climbing.

UNIT VI. Advanced LISP: Macros, lexical scoping, lexical closures, special variables, dynamic scoping, multiple values. Consing, destructive functions, and garbage collection. Interning symbols. Caching. Delaying computation. Avoiding unnecessary consing.

Internal (CA) Evaluation: Practical Note Book (15 marks), Two Experiments (10 marks), Viva-voce (5 marks).

ESE Evaluation: Two Experiments (10 marks), Viva-voce (10 marks).

References/ Suggested Readings:

1. Elaine Rich, Kevin Knight, Shivashankar B Nair, Artificial Intelligence, McGraw Hill Edition.
2. Russell Stuart Jonathan and Norvig Peter, Artificial Intelligence: A Modern Approach, Prentice Hall.
3. Dan W. Patterson, Introduction to Artificial Intelligence and Expert Systems, PHI.

Course Name: Basics of Computer Graphics
Course Code: BSCPCOSDSE602

Course Type: **DSE (Theoretical & Practical)**	Course Details: **DSEC-1(2) or DSEC-2(2) or DSEC-3(2)**		L-T-P: 4 - **0** - **4**	
			CA Marks		ESE Marks	
Credit: **6**	Full Marks: **50**	Practical	Theoretical	Practical	Theoretical
		30	**10**	20	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):

1. *Acquire familiarity with the concepts and relevant mathematics of computer graphics.*
2. *Ability to implement various algorithms to scan, convert the basic geometrical primitives, transformations, area filling, clipping.*
3. *Describe the importance of viewing and projections.*
4. *Ability to design basic graphics application programs.*
5. *Be able to design applications that display graphic images to given specifications.*

Course Content:

Theory

UNIT I. Application Areas of Computer Graphics, Overview of Graphics Systems and Devices. Points and Lines, Line Drawing Algorithms, Mid-Point Circle and Ellipse Algorithms. Filled Area Primitives, Polygon Filling Algorithms. Curve Generation: Bezier and B-Spline Curves.

UNIT II. 2-D Geometrical Transforms: Translation, Scaling, Rotation, Reflection and Shear Transformations Composite Transforms, Transformations between Coordinate Systems. 2-D Viewing: The Viewing Pipeline, Viewing Coordinate Reference Frame, Window to Viewport Coordinate Transformation, Viewing Functions.

UNIT III. Line Clipping Algorithms- Cohen-Sutherland and Cyrus Beck Line Clipping Algorithms, Sutherland–Hodgeman Polygon Clipping Algorithm. 3-D Object Representation: Polygon Surfaces, Quadric Surfaces, Spline Representation

UNIT IV. 3-D Geometric Transformations: Translation, Rotation, Scaling, Reflection and Shear Transformations, Composite Transformations, 3-D Viewing: Viewing Pipeline, Viewing Coordinates, View Volume, General Projection Transforms and Clipping.

UNIT V. Visible Surface Detection Methods: Classification, Back -Face Detection, DepthBuffer, Scanline, Depth Sorting, BSP-Tree Methods, Area Sub-Division and Octree Methods Illumination Models and Surface Rendering Methods: Basic Illumination Models, Polygon Rendering Methods.

Practical

The students are required to create interactive graphics applications in C using graphics application programming interfaces and demonstrate geometrical transformations. The lab material includes implementation of line drawings, circle drawing, ellipse drawing as well as different geometrical transformations.

Experiment 1: Line Drawing Using DDA and Bresenham

Experiment 2: Circle Drawing Using Midpoint Algorithm .

Experiment 3: Ellipse Drawing Using Midpoint Algorithm.

Experiment 4: Line Clipping Algorithms- Cohen-Sutherland and Cyrus Beck.

Experiment 5: Polygon Filling Algorithms.

Experiment 6: Performing the basic 2D transformations such as translation, Scaling, Rotation, shearing and reflection for a given 2D object.

Internal (CA) Evaluation: Practical Note Book (15 marks), One Experiment (10 marks), Viva-voce (5 marks).

ESE Evaluation: One Experiment (10 marks), Viva-voce (10 marks).

References/ Suggested Readings:

1. Donald Hearn and M. Pauline Baker, Computer Graphics with Open GL, Prentice Hall.
2. R. K Maurya, Computer Graphics with Virtual Reality, Wiley.
3. Foley, Van Dam, Feiner and Hughes, Computer Graphics Principles & practice, Pearson Education.
4. D. P. Mukherjee, Fundamentals of Computer Graphics and Multimedia, PHI.

Course Name: Basics of Mobile Application Development

Course Code: BSCPCOSSE601

Course Type: SE (Theoretical)	Course Details: **SEC-4**		L-T-P: **4 - 0 - 0**	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		---	**10**	---	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):
1. *To understand Android platform and its architecture.*
2. *To learn about mobile devices types and different modern mobile operating systems.*
3. *To learn activity creation and Android User Interface designing.*

4. *To learn basics of Intent, Broadcast and Internet services.*
5. *To learn about different wireless mobile data transmission standards.*
6. *To understand and learn how to integrate basic phone features, multimedia, camera and Location based services in Android Application.*
7. *To learn about different systems for mobile application development, deployment and distribution in Mobile market place (Android, IoS).*
8. *To understand and carry out functional test strategies for mobile applications.*

Course Content:

Theory

UNIT I. Introduction: What is Android, Android Versions and its Feature Set, Various Android Devices on the Market, Android Market Application Store, Android Development Environment System Requirements, Android SDK, Installing Java, and ADT bundle - Eclipse Integrated Development Environment (IDE), Creating Android Virtual Devices (AVDs).

UNIT II. Android Architecture Overview and Application: Android Software Stack, The Linux Kernel, Android Runtime - Dalvik Virtual Machine, Android Runtime – Core Libraries, Dalvik VM Specific Libraries, Java Interoperability Libraries, Android Libraries, Application Framework, Creating a New Android Project ,Defining the Project Name and SDK Settings, Project Configuration Settings, Configuring the Launcher Icon, Creating an Activity, Running the Application in the AVD, Stopping a Running Application, Modifying the Example Application, Reviewing the Layout and Resource Files.

UNIT III. Android Software Development Platform and Framework: Understanding Java SE and the Dalvik Virtual Machine, The Directory Structure of an Android Project, Common Default Resources Folders, The Values Folder, Leveraging Android XML, Screen Sizes , Launching Mobile Application: The AndroidManifest.xml File, Android Application Components, Android Activities: Defining the UI, Android Services: Processing in the Background, Broadcast Receivers: Announcements and Notifications Content Providers: Data Management, Android Intent Objects: Messaging for Components, Android Manifest XML: Declaring Your Components.

UNIT IV. Understanding Android User Interfaces, Views and Layouts: Designing for Different Android Devices, Views and View Groups, Android Layout Managers, The View Hierarchy, Designing an Android User Interface using the Graphical Layout Tool Displaying Text with TextView, Retrieving Data from Users, Using Buttons, Check Boxes and Radio Groups, Getting Dates and Times from Users, Using Indicators to Display Data to Users, Adjusting Progress with Seek Bar, Working with Menus using views, Gallery, Image Switcher, Grid View, and Image View views to display images.

UNIT V. Databases, Intents, Location-based Services: Saving and Loading Files, SQLite Databases, Android Database Design, Exposing Access to a Data Source through a Content Provider, Content Provider Registration, Native Content Providers Intents and Intent Filters: Intent Overview, Implicit Intents, Creating the Implicit Intent Example Project, Explicit Intents, Creating the Explicit Intent Example Application, Intents with Activities, Intents with Broadcast Receivers.

UNIT VI. Sending SMS Messages Programmatically, Getting Feedback after Sending the Message Sending SMS Messages Using Intent Receiving, sending email, Introduction to location-based service, configuring the Android Emulator for Location -Based Services, Geocoding and Map-Based Activities.

References/ Suggested Readings:

1. Harwani, Android Programming Unleashed, SAMS.
2. Richard Rodger, Beginning Mobile Application Development in the Cloud, Wrox.

Course Name: Basics of Cloud Computing

Course Code: BSCPCOSSE602

Course Type: **SE (Theoretical)**	Course Details: **SEC-4**		L-T-P: **4 - 0 - 0**	
Credit: **4**	Full Marks: **50**	CA Marks		ESE Marks	
		Practical	Theoretical	Practical	Theoretical
		---	**10**	---	**40**

Course Learning Outcomes:

(After the completion of course, the students will have ability to):
1. *Analyze the trade-offs between deploying applications in the cloud and over the local infrastructure.*
2. *Compare the advantages and disadvantages of various cloud computing platforms.*
3. *Deploy applications over commercial cloud computing infrastructures such as Amazon Web Services, Windows Azure, and Google AppEngine.*
4. *Program data intensive parallel applications in the cloud.*
5. *Analyze the performance, scalability, and availability of the underlying cloud technologies and software.*
6. *Identify security and privacy issues in cloud computing.*

Course Content:

Theory

UNIT I. Introduction to cloud computing: Definition, characteristics, components, Cloud service provider, the role of networks in Cloud computing, Cloud deployment models-private, public & hybrid, Cloud service models, multitenancy, Cloud economics and benefits, Cloud computing platforms - IaaS: Amazon EC2, PaaS: Google App Engine, Microsoft Azure, SaaS.

UNIT II. Virtualization: Virtualization concepts , Server virtualization, Storage virtualization, Storage services, Network virtualization, Service virtualization, Virtualization management, Virtualization technologies and architectures, virtual machine, Measurement and profiling of virtualized applications.

UNIT III. Data in cloud computing: Relational databases, Cloud file systems: GFS and HDFS, BigTable, HBase and Dynamo. MapReduce and extensions: Parallel computing, the map-Reduce model, Parallel efficiency of MapReduce, Relational operations using Map-Reduce, Enterprise batch processing using MapReduce.

UNIT IV. Cloud security: Cloud security fundamentals, Vulnerability assessment tool for cloud, Privacy and Security in cloud. Cloud computing security architecture: General Issues, Trusted Cloud computing, Secure Execution Environments and Communications, Micro - architectures; Identity Management and Access control, Autonomic security, Security challenges : Virtualization security management - virtual threats, VM Security Recommendations.

UNIT V. Issues in cloud computing: Implementing real time application over cloud platform, Issues in Inter-cloud environments, QOS Issues in Cloud, Dependability, data migration, streaming in Cloud. Quality of Service (QoS) monitoring in a Cloud computing environment.

References/ Suggested Readings:

1. Gautam Shroff, Enterprise Cloud Computing,Cambridge publication
2. Ronald Krutz and Russell Dean Vines, Cloud Security, Wiley-India.
3. Dr. Kumar Saurabh, Cloud Computing, Wiley Publication.