Maharashtra Education Society

Abasaheb Garware College

(Autonomous)

(Savitribai Phule Pune University)

Three Year B.Sc. Degree Program in Computer Science

(Faculty of Science and Technology)

Syllabi under Autonomy

S.Y.B.Sc. (Computer Science)

Choice Based Credit System Syllabus

To be implemented from Academic Year 2023-2024

Structure of the Course: B.Sc. (Computer Science)

Year	Semester	Course Type	Course Code	Course Title	Remark	Credit	No. of Lectures /Practical to be conducted
2	III	CC-IX	USCS-231	Data Structures using C		2	36
			USCS-232	Relational Database Management Systems		2	36
			USCSP-233	Computer Science Laboratory		2	12
		CC-X	USCSMT-231	Groups and Coding Theory		2	36
			USCSMT-232	Numerical Techniques		2	36
			USCSMTP-233	Mathematics Laboratory		2	12
		CC-XI	USCSEL-231	Microcontroller Architecture & Programming		2	36
			USCSEL-232	Digital Communication and Networking		2	36
			USCSELP-233	Electronics Laboratory		2	12
		AECC-I	UEVS-231	Environmental Science - I		2	36
		AECC-II	USLGA-231	Language Communication – I		2	36
	IV	CC-XII	USCS-241	Object Oriented Programming using C++		2	36
			USCS-242	Software Engineering		2	36
			USCSP-243	Computer Science Laboratory		2	12
		CC-XIII	USCSMT-241	Computational Geometry		2	36
			USCSMT-242	Operations Research		2	36
			USCSMTP-243	Mathematics Laboratory		2	12
		CC-XIV	USCSEL-241	Embedded System Design		2	36
			USCSEL-242	Wireless Communication and Internet of Things		2	36
			USCSELP-243	Electronics Laboratory		2	12
		AECC-III	UEVS-241	Environmental Science - II		2	36
		AECC-IV	USLGA-241	Language Communication - II		2	36

SECOND YEAR SEMESTER-I

Course Code and Title: USCS-231 Data Structures using C

Lectures: 36 (Credits-2)

Prerequisites:

- Basic knowledge of algorithms and problem solving
- Knowledge of C Programming Language

Course Objectives:

1. To understand analysis of algorithms.
2. To learn different array representation methods and uses.
3. To understand different types of linked list.
4. To learn use of stack.
5. To learn concept of queue with its operations.
6. To acquire knowledge of nonlinear data structures.

Learning Outcomes: On completion of this course, students will be able to:

1. Analyze the algorithms on the scale of their performance.
2. Develop searching and sorting techniques to solve real world computing problems.
3. Apply linked list data structure for developing applications.
4. Implement various applications of stack.
5. Use queue operations for various applications.
6. Understand tree and graph operations with its applications.

Unit 1: Introduction to Data Structures and Algorithm Analysis 02

1.1 Introduction
- 1.1.1 Need of Data Structure
- 1.1.2 Data object, Data Structure, Abstract Data Type (ADT)
- 1.1.3 Types of Data Structures

1.2 Algorithm analysis
- 1.2.1 Space complexity, time complexity and examples-linear, logarithmic, quadratic etc.
- 1.2.2 Best, Worst, Average case analysis, Asymptotic notations (Big O, Omega Ω, Theta θ) Problems on time complexity calculation.

Unit 2: Array as a Data Structure 08

2.1 ADT of array, Operations on Arrays

2.2 Applications of Arrays
- 2.2.1 Searching
  - 2.2.1.1 Various sequential search methods - Sentinel search, Probability search, Ordered list search
  - 2.2.1.2 Binary Search

2.2.1.3 Comparison of searching methods

2.2.2 Sorting

2.2.2.1 Terminology, Internal, External, Stable, In-place Sorting

2.2.2.2 Comparison Based Sorting methods - Bubble Sort, Insertion Sort, Selection Sort

2.2.2.3 Non-Comparison Based Sorting algorithms - Counting Sort, Radix Sort

2.2.2.4 Divide and Conquer strategy - Merge Sort, Quick Sort

2.2.2.5 Complexity analysis of sorting methods.

Unit 3: Linked List **08**

6.1 List as a Data Structure, comparison with array.

6.2 Dynamic implementation of Linked List, internal and external pointers

6.3 Types of Linked List

6.3.1 Singly linked list

6.3.2 Doubly linked list

6.3.3 Circular list

6.3.4 Doubly circular linked list

6.4 Operations on Linked List - create, traverse, insert, delete, search, sort, reverse, concatenate, merge, time complexity of operations

6.5 Applications of Linked List – Polynomial representation, Addition of two polynomials

Unit 4: Stack **06**

4.1 Introduction

4.2 Operations on Stack – init(), push(), pop(), isEmpty(), isFull(), peek(), time complexity of all the operations

4.3 Implementation of Stack - Static and Dynamic, comparison between the two implementations

4.4 Applications of stack

4.4.1 Function call and recursion

4.4.2 String reversal, palindrome checking

4.4.3 Expression types - infix, prefix and postfix, expression conversion and evaluation (implementation of infix to postfix and evaluation of postfix expression)

4.4.4 Backtracking strategy - 4 queens problem

Unit 5: Queue **06**

5.1 Introduction

5.2 Operations on Queue - init(), enqueue(), dequeue(), isEmpty(), isFull(), peek(), time complexity of all the operation

5.3 Implementation - Static and Dynamic, comparison between the two implementations

5.4 Types of Queue - Linear Queue, Circular Queue, Priority Queue, Double Ended Queue

5.5 Applications of queue

6.5.1 CPU Scheduling in multiprogramming environment

Unit 6: Non-linear data Structures **06**

6.1 Concept of tree and Terminologies
6.2 Concept and types of Binary trees - skewed tree, strictly binary tree, full binary tree, complete binary tree, expression tree, binary search tree, Heap
6.3 Representation of binary tree (dynamic)
6.4 Traversal methods of Binary tree – preorder, inorder, postorder (Recursive implementation)
6.5 Operations on binary search tree - create, insert node, search a value, copy a tree, mirroring of tree
6.6 Applications of tree
6.7 Concept of graph and terminologies
6.8 Representation of Graph – Adjacency matrix, Adjacency list, Inverse Adjacency list, Adjacency multilist
6.9 Graph Traversals – Breadth First Search and Depth First Search
6.10 Applications of graph

Reference Books:

1. Classic Data Structures - D. Samanta, Prentice Hall India Pvt. Ltd.
2. Fundamentals of Data Structures in C - Ellis Horowitz, Sartaj Sahni, Susan Anderson Freed, 2nd Edition, Universities Press.
3. Data Structures using C and C++ - Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum, Pearson Education
4. Data Structures: A Pseudo code approach with C, Richard Gilberg, Behrouz A. Forouzan, Cengage Learning.
5. Introduction to Data Structures in C - Ashok Kamthane, Pearson Education
6. Algorithms and Data Structures - Niklaus Wirth, Pearson Education

Course Code and Title: USCS-232 Relational Database Management System

Lectures: 36 (Credits-2)

Prerequisites:

- Basic Knowledge of RDBMS
- Knowledge of SQL Queries
- Basics of relational database design

Course Objectives:

1. To understand the concept, need and procedure for Normalization
2. To teach procedural extension of SQL
3. To be familiar with the basic concepts of transaction and concurrency control
4. To understand how recovery is done from crash
5. To understand data security and its importance

Learning Outcomes: On completion of this course, students will be able to:

1. Chek for Normal forms, convert data in required Normal form.
2. Use database techniques such as SQL & PL/SQL
3. Understand transaction Management in relational database System.
4. Understand how recovery from crash is handled in DBMS
5. Learn Database Security mechanisms.

Unit 1: Relational Database Design 05

1.1 Introduction to Relational-Database Design (undesirable properties of a RDB design)

1.2 Concept of Decomposition

1.3 Desirable Properties of Decomposition (Lossless join, Lossy join, Dependency Preservation)

1.4 Concept of normalization, Normal Forms (1NF,2NF and 3NF, BCNF), Examples

1.5 Brief History of NoSQL Databases

1.6 NoSQL Database Features

1.7 Difference between RDBMS and NoSQL

1.8 Need of NoSQL

1.9 Vertical and Horizontal Scaling (Scale-Up, Scale-Out)

Unit 2: Relational Database Design Using PLSQL 12

2.1 Introduction to PLSQL

2.2 PL/PgSQL: Datatypes, Language structure

2.3 Controlling the program flow, conditional statements, loops

2.4 Stored Functions

2.5 Handling Errors and Exceptions

2.6 Cursors

2.7 Triggers

2.8 Views

Unit 3: Transaction Concepts and concurrency control **08**

3.1 Describe a transaction, properties of transaction, state of the transaction.

3.2 Executing transactions concurrently, Anomalies due to Interleaved Execution.

3.3 Schedules, types of schedules, concept of Serializability, Precedence graph for Serializability.

3.4 Ensuring Serializability by locks, different lock modes, 2PL and its variations.

3.5 Concurrency control without Locking

3.5.1 Optimistic Concurrency Control

3.5.2 Timestamp Based Concurrency, Thomas Write Rule.

3.5.3 Multiversion Concurrency Control

3.6 Deadlock

3.7 Deadlock handling

3.7.1 Deadlock Prevention (wait-die, wound-wait)

3.7.2 Deadlock Detection and Recovery (Wait for graph)

Unit 4: Crash Recovery **06**

4.1 Failure classification

4.2 Buffer Management: Stealing Frames and Forcing Pages

4.3 Recovery related steps during Normal Execution

4.4 The Log, other Recovery-related structures.

4.5 The Write-Ahead Log Protocol

4.6 Checkpoints

4.7 Recovering from a System Crash

4.7.1 Analysis Phase

4.7.2 Redo Phase

4.7.3 Undo Phase

4.8 Database backup and recovery from catastrophic failure

4.9 Concept of Importing and Exporting data from other databases

Unit 5: Database Integrity and Security Concepts **05**

5.1 Introduction to Database security

5.2 Access Control

5.3 Methods for database security

5.3.1 Discretionary access control method
5.3.2 Mandatory access control
5.3.3 Security for Internet Application
5.4 Additional Issues related to Security
5.4.1 Role of Database Administrator
5.4.2 Security in Statistical Databases

Reference Books:

1. Database Management Systems by Raghu Ramakrishnan, Mcgraw-hill higher Education publication
2. Database System Concepts by Henry F. Korth, Abraham Silberschatz, Tata McGraw-Hill Education publication
3. Fundamentals of Database Systems by Elmasri and Navathe, Pearson publication
4. Beginning Databases with PostgreSQL: From Novice to Professional by Richard Stones, Neil Matthew, Apress publication
5. Practical PostgreSQL By Joshua D. Drake, John C Worsley O'Reilly publication

Course Code and Title: USCSP-233 Computer Science Laboratory
No. of Sessions – 12 (Credits – 2)

Assignments of Data Structures using C:

1. Assignment on array: Matrix operations
2. Searching algorithms
   a. Sequential Search
   b. Binary Search
3. Sorting algorithms
   a. Bubble, Insertion Selection Sort
   b. Quick, Merge Sort
   c. Count sort
4. Operations on Stack
5. Applications of Stack – palindrome, Conversion to postfix and evaluation of postfix
6. Operations on Queue and Implementation of priority queue
7. Operations on Singly linked list
8. Operations on Doubly linked list
9. Tree Traversal
10. Binary Search tree – create(), insert(), searc()

Assignments of Relational Database Management System:

1. Stored Function
   a. A Simple Stored Function
   b. A Stored Function that returns a value
   c. A Stored Function recursive
2. Cursors
   a. A Simple Cursor
   b. A Parameterize Cursor
3. Exception Handling
   a. Simple Exception- Raise Debug Level Messages
   b. Simple Exception- Raise Notice Level Messages
   c. Simple Exception- Raise Exception Level Messages
4. Triggers
   a. Before Triggers (insert, update, delete) (row level and statement level)
   b. After Triggers (insert, update, delete) (row level and statement level)
5. Views
   a. Creating Views on tables, and writing queries on the views.

Course Code and Title: USCSMT231 Groups and Coding Theory
Lectures: 36 (Credits-2)

Unit 1: Integers **05**

1.1 Division Algorithm (without Proof)
1.2 G.C.D. using division algorithm and expressing it as linear combination
1.3 Euclid's lemma
1.4 Equivalence relation (revision), Congruence relation on set of integers, Equivalence class partition

Unit 2: Groups **03**

2.1 Binary Operation
2.2 Group: Definition and Examples
2.3 Elementary Properties of Groups

Unit 3: Finite Groups and Subgroups **10**

3.1 Order of a group, order of an element
3.2 Examples (Zn, +) and (U(n),∗)
3.3 Subgroup definition, Finite subgroup test, subgroups of Zn
3.4 Generator, cyclic group, finding generators of Zn (Corollary 3,4 without proof)
3.5 Permutation group, definition, composition of two permutations, representation as product of disjoint cycles, inverse and order of a permutation, even/ odd permutation
3.6 Cossets: Definition, Examples and Properties, Lagrange Theorem (without Proof)

Unit 4: Groups and Coding Theory **18**

4.1 Coding of Binary Information and Error detection
4.2 Decoding and Error Correction
4.3 Public Key Cryptography

Reference Books:-

1. Contemporary Abstract Algebra by J. A, Gallian (Seventh Edition)
   Unit 1: Chapter 0, Unit 2: Chapter 2, Unit 3: Chapter 3 ,4, 5 and 7
2. Discrete Mathematical Structures By Bernard Kolman, Robert C. Busby and Sharon Ross (6^{th} Edition) Pearson Education Publication
   Unit 4: Chapter 11

Course Code and Title: USCSMT232 Numerical Techniques
Lectures: 36 (Credits-2)

Unit 1: Algebraic and Transcendental Equation **04**

1.1 Introduction to Errors
1.2 False Position Method
1.3 Newton-Raphson Method

Unit 2: Calculus of Finite Differences and Interpolation **16**

2.1 Differences
  2.1.1 Forward Differences
  2.1.2 Backward Differences
  2.1.3 Central Differences
  2.1.4 Other Differences (δ, μ operators)
2.2 Properties of Operators
2.3 Relation between Operators
2.4 Newton's Gregory Formula for Forward Interpolation
2.5 Newton's Gregory Formula for Backward Interpolation
2.6 Lagrange's Interpolation Formula
2.7 Divided Difference
2.8 Newton's Divided Difference Formula

Unit 3: Numerical Integration **08**

3.1 General Quadrature Formula
3.2 Trapezoidal Rule
3.3 Simpson's one-Third Rule
3.4 Simpson's Three-Eight Rule

Unit 4: Numerical Solution of Ordinary Differential Equation **08**

4.1 Euler's Method
4.2 Euler's Modified Method
4.3 Runge-Kutta Methods

Text Book:-

1. A textbook of Computer Based Numerical and Statistical Techniques, by A. K. Jaiswal and Anju Khandelwal. New Age International Publishers.
   Chapter 1:2.1, 2.5, 2.7
   Chapter 2:3.1, 3.2, 3.4, 3.5,4.1, 4.2, 4.3, 5.1, 5.2, 5.4, 5.5
   Chapter 3:6.1, 6.3, 6.4, 6.5, 6.6, 6.7
   Chapter 4:7.1, 7.4, 7.5, 7.6

Reference Books:-

1. S.S. Sastry; Introductory Methods of Numerical Analysis, 3rd edition, Prentice Hall of India, 1999.
2. H.C. Saxena; Finite differences and Numerical Analysis, S. Chand and Company.
3. K.E. Atkinson; An Introduction to Numerical Analysis, Wiley Publications.
4. Balguruswamy; Numerical Analysis.

Course Code and Title: USCSMTP 233 – Mathematics Practical
No. of Sessions – 12 (Credits – 2)

Python Programming Language Syllabus

1. Introduction to Python, Python Data Types I (Unit 1)
2. Python Data Types II (Unit 2)
3. Control statements in Python I (Unit 3- 3.1, 3.2)
4. Control statements in Python II (Unit 3- 3.3)
5. Application : Matrices (Unit 5 – 5.1-5.3)
6. Application : Determinants, system of Linear Equations (Unit 5- 5.4, 5.5)
7. Application : System of equations (Unit 5- 5.5)
8. Application : Eigenvalues, Eigenvectors (Unit 5 – 5.6)
9. Application : Eigenvalues, Eigenvectors (Unit 5 – 5.6)
10. Application : Roots of equations (Unit 6 – 6.1)
11. Application : Numerical integration (Unit 1 – 6.2.1, 6.2.2)
12. Application : Numerical integration (Unit 1 – 6.2.3)

Text Books:-

1. Downey, A. et al., How to think like a Computer Scientist: Learning with Python, John Wiley, 2015.
   Sections: 1, 2, 3
2. Robert Johansson, Introduction to Scientific Computing in Python Section: 4

Reference Books:-

1. Lambert K. A., Fundamentals of Python - First Programs, Cengage Learning India, 2015.
2. Guzdial, M. J., Introduction to Computing and Programming in Python, Pearson India.
3. Perkovic, L., Introduction to Computing Using Python, 2/e, John Wiley, 2015.
4. Zelle, J., Python Programming: An Introduction to Computer Science, Franklin, Beedle & Associates Inc.
5. Sandro Tosi, Matplotlib for Python Developers, Packt Publishing Ltd.(2009)

Course Code and Title: USCSEL-231 Microcontroller Architecture & Programming

No. of Lectures 36 (Credits 2)

Objectives:

1. To study the basics of 8051microcontroller
2. To study the Programming of8051microcontroller
3. To study the interfacing techniques of 8051microcontroller
4. To design different application circuits using 8051microcontroller

Course Outcomes: On completion of the course, student will be able

1. To write programs for 8051 microcontroller
2. To interface I/O peripherals to 8051 microcontroller
3. To design small microcontroller based projects

UNIT- 1: Basics of Microcontroller & Intel 8051 architecture 08

1.1 Introduction to microcontrollers
1.2 Difference in controller and processor
1.3 Architecture of 8051
1.4 Internal block diagram
1.5 Internal RAM organization
1.6 SFRS
1.7 Pin functions of 8051
1.8 I/O port structure & Operation
1.9 External Memory Interface

UNIT-2: Programming model of 8051 10

2.1 Instruction classification
2.2 Instruction set
2.3 Addressing Modes - Immediate, register, direct, indirect and relative
2.4 Assembler directives (ORG, END)
2.5 Features with examples
2.6 I/O Bit & Byte programming using assembly language for LED and seven segment display (SSD) interfacing
2.7 Introduction to 8051 programming in C

UNIT- 3: Timer /Counter, Interrupts 10

3.1 Timer / counter: TMOD, TCON, SCON, SBUF, PCON Registers
3.2 Timer modes
3.3 Programming for time delay using mode 1 and mode 2
3.4 Interrupts

3.4.1 Introduction to interrupt
3.4.2 Interrupt types and their vector addresses
3.4.3 Interrupt enable register and interrupt priority register (IE, IP)

UNIT- 4: Interfacing, Serial Communication **08**

4.1 Programming of serial port without interrupt
4.2 Serial Communication - Synchronous and asynchronous serial communication
4.3 Use of timer to select baud rate for serial communication
4.4 Interfacing - ADC, DAC, LCD, stepper motor

Reference Books :

1. 8051 microcontroller and Embedded system using assembly and C : Mazidi and McKinley, Pearson publications
2. The 8051 microcontroller – Architecture, programming and applications: K.Uma Rao and Andhe Pallavi, Pearson publications

Course Code and Title: USCSEL-232 Digital Communication and Networking

No. of Lectures 36 (Credits 2)

Objectives:

1. To introduce to various aspects of data communication system
2. To introduce different digital modulation schemes
3. To identify the need of data coding and error detection/correction mechanism.
4. To study bandwidth utilization techniques: multiplexing and spectrum spreading
5. To understand data link layer protocol: Media Access Control

Course Outcomes: On completion of the course, student will be able to

1. Define and explain terminologies of data communication
2. Understand the impact and limitations of various digital modulation techniques
3. Acknowledge the need of spread spectrum schemes.
4. Identify functions of data link layer and network layer while accessing communicationlink
5. Choose appropriate and advanced techniques to build the computer network

UNIT 1: Introduction to Electronic Communication **09**

1.1 Introduction to Communication
- 1.1.1 Elements of Communication system
- 1.1.2 Types of noise sources
- 1.1.3 Electromagnetic spectrum
- 1.1.4 Signal and channel bandwidth

1.2 Types of communication
- 1.2.1 Simplex
- 1.2.2 Half duplex
- 1.2.3 Full duplex
- 1.2.4 Baseband and broadband

1.3 Serial communication
- 1.3.1 Asynchronous and synchronous,

1.4 Information Theory
- 1.4.1 Information entropy
- 1.4.2 Rate of information (data rate, baud rate)
- 1.4.3 Channel capacity
- 1.4.4 Signal to noise ratio
- 1.4.5 Noise Figure
- 1.4.6 Shannon theorem

1.5 Error handling codes: Necessity, Hamming code, CRC

UNIT 2: Modulation and Demodulation **05**

2.1 Introduction to modulation and demodulation

2.1.1 Concept and need of modulation and demodulation

2.2 Digital Modulation techniques

2.2.1 Pulse Code Modulation (PCM)

2.2.2 FSK

2.2.3 QPSK

2.2.4 QAM

UNIT 3: Multiplexing, Spectrum Spreading and Media Access Control **12**

3.1 Multiplexing techniques

3.1.1 Frequency division multiplexing

3.1.2 Wavelength division multiplexing

3.1.3 Time division multiplexing

3.2 Spread Spectrum techniques

3.2.1 Frequency hopping Spread Spectrum (FHSS)

3.2.2 Direct SequenceSpread Spectrum (DSSS)

3.3 Media Access Control (MAC)

3.3.1 Random Access Protocol - CSMA, CSMA/CD, CSMA/CA

3.3.2 Controlled Access Protocols - Reservation, Polling, Token passing

UNIT 4: Computer Networking **10**

4.1 Introduction to computer networks

4.2 Types of networks - LAN, MAN, WAN, Wireless networks, Switching, Internet,

4.3 Network topology - point to point, Star, Ring, Bus, Mesh, Tree, Daisy Chain, Hybrid

4.4 Network devices - Repeater, Switch, Networking cables, Router, Bridge, Hub, Brouter,Gateway. Wired LANs

4.5 Ethernet - Ethernet protocol, standard Ethernet, 100 MBPS Ethernet, Gigabit Ethernet, 10Gigabit Ethernet,

4.6 Computer network model - OSI and TCP/IP.

Reference Books:

1.Communication Electronics: Principles and Applications, Frenzel, Tata Mc Graw Hillpublication, 5th edition.

2.Data Communication and Networking, Forouzan, Mc Graw Hill publication, 5th edition

3.Computer Networks, Tanenbaum, pHI publication, 5th edition

Course Code and Title: USCSELP-233 Electronics Laboratory
No. of sessions 12 (Credits 2)

Objectives:

1. To get hands on training of Embedded C
2. To study experimentally interfacing of microcontroller
3. To design, build and test modulator and demodulators of digital communication
4. To build and test experimentally various techniques of wired communication
5. To develop practical skills of network setup

Course Outcomes : On completion of the course, student will be able

1. To design and build his/her own microcontroller based projects.
2. To acquire skills of Embedded C programming
3. To know multiplexing and modulation techniques useful in developing wireless application
4. Do build and test own network and do settings.

Guidelines for Practical:

- Practical batch size : 12
- Minimum no of Practical to be performed : 10
- At least five practical from each Group
- Electronics lab should have set up for embedded programming (Computers and microcontroller target and interfacing boards)

Group A:

1. Interfacing of thumbwheel & seven segment display to 8051 microcontroller
2. Traffic light controller using 8051 microcontroller
3. Waveform generation using DAC Interface to 8051Microcontroller.
4. Speed Control of stepper motor using 8051 microcontroller

Group B:

1. Study of 3 or 4 Bit Pulse Code Modulation technique
2. Study of Frequency Shift Keying
3. Study of Time Division Multiplexing
4. Study of Frequency Division Multiplexing
5. Study of Error detection and correction by using Hamming Code technique

SECOND YEAR SEMESTER-II

Course Code and Title: USCS-241 Object Oriented Programming using C++

Lectures: 36 (Credits-2)

Prerequisites: Knowledge of C Programming Language

Course Objectives:

1. To learn principles of Object-Oriented Programming (OOP).
2. To learn operators, type casting, reference variable
3. To understand concept of functions, default arguments
4. To acquire the knowledge of function and operator overloading.
5. To learn concept of Inheritance, abstract class

Learning Outcomes: On completion of this course, students will be able to:

1. Understand object-oriented concepts such as data abstraction, encapsulation, inheritance, dynamic binding and polymorphism.
2. Make use of this pointer and access specifiers.
3. Apply new and delete operator.
4. Understand concept of polymorphism.
5. Illustrate concept of inheritance, virtual function, pure virtual function.

Unit 1: Object oriented concepts **02**

1.1 Procedure-oriented programming Vs Object-oriented programming
1.2 Classes and objects
1.3 Abstraction
1.4 Inheritance
1.5 Polymorphism
1.6 Encapsulation

Unit 2: Introduction to 'C++' programming **08**

2.1 Features of C++
2.2 Data Types and Operators
2.3 Type casting in C++
2.4 Reference variables, this pointer
2.5 Usage of namespace, Managing Console I/O, Usage of Manipulators
2.6 Access specifiers
2.7 Defining data members and member functions
2.8 Array of objects

Unit 3: Function in C++ **08**

3.1 Constructor, types of constructors, Destructor
3.2 Memory allocation (new and delete)
3.3 Call by reference, Return by reference
3.4 Default arguments
3.5 Inline function
3.6 Static class members
3.7 Friend functions

Unit 4: Function and Operator overloading **08**

4.1 Function overloading
4.2 Overloading unary operators and binary operators (with member function and with friend function)
4.3 Overloading insertion and extraction operator

Unit 5: Inheritance **10**

5.1 Inheritance and its types
5.2 Constructor and destructor in derived class
5.3 Virtual base classes and abstract base classes
5.4 Function overriding
5.5 Virtual functions and pure virtual function

Reference Books:

1. Object Oriented Programming (C++) Balaguruswamy, McGraw Hill Education; Seventh edition
2. The Complete Reference C++ by Herbert Schildt, McGraw Hill Education; 4 edition
3. Mastering C++ by Venugopal, T Ravishankar, McGraw Hill Education; 2 edition A Structured Programming Approach Using C, Behrouz A. Forouzan, Richard F. Gilberg, Cengage Learning India

Course Code and Title: USCS-242 Software Engineering
Lectures: 36 (Credits-2)

Prerequisites: None

Course Objectives:

1. To enhance the knowledge of methods and processes involved in software design and development with object-oriented concepts.
2. To get introduced to the concepts of agile software development and methodologies like scrum and extreme programming.
3. To learn concepts of Requirement Analysis, building of SRS.
4. To learn basic UML and construction of Class Diagram, Object Diagram.
5. To understand Behavioral and Architectural Modeling using UML diagrams.

Learning Outcomes: On completion of this course, students will be able to:

1. Design a system using software engineering principles.
2. Get basic knowledge of Agile Software Development.
3. Perform requirement analysis and build SRS.
4. Construct UML diagrams like Class Diagram and Object Diagram.
5. Apply Behavioral and Architectural Modeling techniques and draw different UML diagrams like Use case, Activity, State, Sequence, Component and Deployment.

Unit 1: Introduction To Software Engineering and Process Models 06

1.1 Definition of Software
1.2 Nature of Software Engineering
1.3 Changing nature of software
1.4 Software Process
  1.4.1 The Process Framework
  1.4.2 Umbrella Activities
  1.4.3 Process Adaptation
1.5 Generic Process Model
1.6 Prescriptive Process Models
  1.6.1 The Waterfall Model
  1.6.2 Incremental Process Models
  1.6.3 Evolutionary Process Models
  1.6.4 Concurrent Models
  1.6.5 The Unified Process

Unit 2: Agile Development 08

2.1 What is Agility?
2.2 Agile Process
  2.2.1 Agility Principles
  2.2.2 The Politics Of Agile Development
  2.2.3 Human Factors

2.3 Extreme Programming(XP)
- 2.3.1 XP Values
- 2.3.2 XP Process
- 2.3.3 Industrial XP

2.4 Adaptive Software Development(ASD)

2.5 Scrum

2.6 Dynamic System Development Model (DSDM)

2.7 Agile Unified Process (AUP)

Unit 3: Requirements Analysis **06**

3.1 Requirements Engineering
3.2 Establishing Groundwork
3.3 Requirement Elicitation,
3.4 Software requirement specification (SRS)
3.5 Building the Analysis Model
- 3.3.1 Elements of the Analysis Model
- 3.3.2 Analysis Patterns
- 3.3.3 Agile Requirements Engineering

3.6 Negotiating Requirements
3.7 Requirements Monitoring
3.8 Validating Requirements

Unit 4: Introduction of UML and Basic and Advanced Structural Modeling **08**

4.1 Overview of UML,
4.2 Conceptual Model of UML.
4.3 Class Diagram, Advanced classes
4.4 Advanced Relationship
4.5 Interface
4.6 Types and Roles
4.7 Packages
4.8 Object Diagram

Unit 5: Basic Behavioral and Architectural Modeling **08**

5.1 Use case diagram
5.2 Activity diagram
5.3 State diagram
5.4 Sequence diagram
5.5 Component Diagram
5.6 Deployment Diagram

Reference Books:

1. Software Engineering: A Practitioner's Approach- Roger S. Pressman, McGraw hill International Editions 2010(Eighth Edition)
2. Grady Booch, James Rambaugh, The Unified Modeling Language User/Reference Guide, Second Edition, Publisher: Addison Wesley, ISBN-13: 978-0321267979
3. Ivar Jacobson, Object Oriented Software Engineering, First Edition, Addison Wesley, ISBN-13: 978-0201544350
4. Mike Kohn, Succeeding with Agile: Software Development Using Scrum, First Edition, Pearson Addison-Wesley Professional, ISBN-13: 978-0321579362
5. Andrew Stellman & Jennifer Greene, Learning Agile understanding scrum, xp, lean, and kanban, First Edition, O'Reilly, ISBN-13 : 978-1449331924

Course Code and Title: USCSP-243 Computer Science Laboratory
No. of Sessions – 12 (Credits – 2)

Assignments of C++

1. Class and Object, Array of objects
2. Functions
   a. Inline function
   b. Static function
3. Overloading
   a. Function overloading
   b. Operator overloading
4. Inheritance, Function overriding
5. Virtual functions, Pure virtual function

Assignments Software Engineering (Mini Project):

1. Prepare detailed statement of problem for the selected mini project
2. Identify suitable process model for the same.
3. Develop Software Requirement Specification for the project.
4. Identify scenarios and develop UML Use case
5. Other artifacts: Class Diagram, activity diagram, sequence diagram, component diagram and any other diagrams as applicable to the project.

Course Code and Title: USCSMT241: Computational Geometry
Lectures: 36 (Credits-2)

Unit 1. Two dimensional transformations: 12

1.1 Introduction.
1.2 Representation of points.
1.3 Transformations and matrices.
1.4 Transformation of points.
1.5 Transformation of straight lines
1.6 Midpoint Transformation
1.7 Transformation of parallel lines
1.8 Transformation of intersecting lines
1.9 Transformation: rotations, reflections, scaling, shearing.
1.10 Combined transformations.
1.11 Transformation of a unit square.
1.12 Solid body transformations.
1.13 Translations and homogeneous coordinates.
1.14 Rotation about an arbitrary point.
1.15 Reflection through an arbitrary line.

Unit 2. Three dimensional transformations: 08

2.1 Introduction.
2.2 Three dimensional – Scaling, shearing, rotation, reflection, translation.
2.3 Multiple transformations.
2.4 Rotation about – an axis parallel to coordinate axes, an arbitrary line
2.5 Reflection through – coordinate planes, planes parallel to coordinate
2.6 Planes, an arbitrary plane

Unit 3. Projection 08

3.1 Orthographic projections.
3.2 Axonometric projections.
3.3 Oblique projections
3.4 Single point perspective projection

Unit 4. Plane and space Curves: 08

4.1 Introduction.
4.2 Curve representation.
4.3 Parametric curves.
4.4 Parametric representation of a circle and generation of circle.
4.5 Bezier Curves – Introduction, definition, properties (without proof),
4.6 Curve fitting (up to n = 3), equation of the curve in matrix form (upto n = 3)

Textbook:

1. D. F. Rogers, J. A. Adams, Mathematical elements for Computer graphics, Mc Graw Hill Intel Edition**.**
   Chapter 1: 2-1 to 2.17
   Chapter 2: 3.1 to 3.10,
   Chapter 3: 3.12 to 3.14

Chapter 4: 4.1, 4.2, 4.5, 5.1, 5.8

Reference books:

1. Computer Graphics with OpenGL, Donald Hearn, M. Pauline Baker, Warren Carothers, Pearson (4th Edition) - Schaum Series, Computer Graphics.

Course Code and Title: USCSMT242: Operations Research

Lectures: 36 (Credits-2)

Unit 1 : Linear Programming Problem I **12**

1.1 Introduction Definition and Examples
1.2 Problem solving using Graphical method
1.3 Theory of Linear Programming, Slack and surplus variables, Standard form of LPP, Some important definitions, Assumptions in LPP, Limitations of Linear programming, Applications of Linear programming, Advantages of Linear programming Techniques
1.4 Simplex method, Big- M-method

Unit 2 : Linear Programming Problem II **08**

2.1 Special cases of LPP :Alternative solution, Unbounded solution, Infeasible solution
2.2 Duality in Linear Programming, Primal to dual conversion, Examples

Unit 3 : Assignment Models **06**

3.1 Assignment Model -Introduction
3.2 Hungerian method for Assignment problem

Unit 4 : Transportation Models **10**

4.1 Introduction, Tabular representation
4.2 Methods of IBFS (North-West rule, Matrix-minima, Vogel's Approximation), Algorithms
4.3 The Optimality Test of Transportation Model (MODI method only)

Text Book:-

1. Operation Research (12 thEdition), by S.D.Sharma.
   Chapter 1: 1.1,1.3-1,1.3-2,1.5,1.6,1.8,1.9,1.10,1.11,1.12,3.1,3.2,3.3,3.4,3.5-4,
   Chapter 2: 3.8-1,3.8-2,5.1-1, 5.2-1,5.3,5.7-1, 5.7-2
   Chapter 3: 9.1, 9.2, 9.4-1, 9.4-2, 9.5, 9.6, 9.7-1, 9.7-2
   Chapter 4: 10.1, 10.2, 10.5, 10.8-1,10.9, 10.10

Reference Books:-

1. Operations Research by H. A. Taha
2. Operations Research by R. Panneerselvam, Prentice Hall of India.
3. Principles of Operations Research by H. M. Wagner, Prentice Hall of India.
4. Operations Research by Gupta and Hira.
5. Operation Research by J.K. Sharma

Course Code and Title: USCSMTP 243 – Mathematics Practical
No. of Sessions – 12 (Credits – 2)

Python Programming Language Syllabus

1. Graph Plotting (Unit 1 – 1.1 – 1.3)
2. Graph Plotting (Unit 1 – 1.4 – 1.7)
3. Application to Computational Geometry (Unit 2 – 2.1)
4. Application to Computational Geometry (Unit 2 – 2.2)
5. Application to Computational Geometry (Unit 2 – 2.3)
6. Study of Graphical aspects of Two dimensional transformation matrix using matplotlib
7. Study of Graphical aspects of Three dimensional transformation matrix using matplotlib
8. Study of Graphical aspects of Three dimensional transformation matrix using matplotlib
9. Study of effect of concatenation of Two dimensional
10. Study of effect of concatenation ofThree dimensional transformations
11. Study of Operational Research in Python (Unit 3.1)
12. Study of Operational Research in Python (Unit 3.2)

1. 2D, 3D Graphs
   1.1. Installation of numpy, matplotlib packages
   1.2. Graphs plotting of functions such as x^2, x^3, $sine$, $cosine$, log, e^x etc.
   1.3. Different formats of graphs.
   1.4. Three-dimensional Points and Lines
   1.5. Three-dimensional Contour Plots
   1.6. Wireframes and Surface Plots
   1.7. Graphs plotting of functions such as$x^2 + y^2$, $sin(x + y)$, $log(x + y)$, $e^{x^2+y^2}$ etc.

2. Computational Geometry
   2.1. Points: The distance between two points, Displaying Points and other geometrical objects, Lines, rays, and line segments, The geometry of line segments, Displaying lines, rays and line segments
   2.2. **Polygon :**Representing polygons in Python, Triangles, Signed area of a triangle
   2.3. Two dimensional rotation and reflection
   2.4. Three dimensional rotation and reflection
   2.5. Generation of Bezier curve with given control points

3. Study of Operational Research in Python
   3.1 Linear Programming in Python
   3.2 Introduction to Simplex Method in Python

Text Books:-

1. JaanKiusalaas, Numerical Methods in Engineering with Python, Cambridge University Press, (2005)
   Sections: 3
2. Robert Johansson, Introduction to Scientific Computing in Python
   Section: 1

3. Jason Brownlee, Basics of Linear Algebra for Machine Learning, Discover the Mathematical Language of Data in Python

Sections: 2

Reference Books:-

1. Lambert K. A., Fundamentals of Python - First Programs, Cengage Learning India, 2015.
2. Guzdial, M. J., Introduction to Computing and Programming in Python, Pearson India.
3. Perkovic**,** L., Introduction to Computing Using Python, 2/e, John Wiley, 2015.
4. Zelle, J., Python Programming: An Introduction to Computer Science, Franklin, Beedle & Associates Inc.
5. Jim Arlow, Interactive Computational Geometry in Python

Course Code and Title: USCSEL-241 Embedded System Design
No. of lectures 36 (Credits 2)

Objectives:

1. To understand the concept of Embedded systems.
2. To study the design flow and available tools for an Embedded system.
3. To understand the implementation of embedded system using firmware and hardware components.
4. To acquire programming skills for the development of Embedded system design.
5. To develop practical skills for designing embedded system Applications.

Course Outcomes: On completion of the course, student will be able

2. To understand the difference between general computing and the Embedded systems.
3. To know the fundamentals of embedded systems.
4. Understand the use of Single board Computer (Such as Raspberry Pi) for an embedded system application.
5. Familiar with the programming environment to develop embedded systems and their interfaces with peripheral devices.
6. To develop familiarity with tools used to develop in an embedded environment.

Unit 1: Introduction to Embedded systems using single board computers (SBC) 08

1.1 Single boards computer block diagram, types
1.2 Comparison of SBC models
1.3 Specifications
1.4 I/O devices (Storage, display, keyboard and mouse)
1.5 Network access devices

Unit 2: Architecture of System on Chip (SOC) 08

3.5 Architecture of SoC,
3.6 Basic version Broad Coprocessor
3.7 Pin Description of Raspberry Pi
3.8 Architectural features
- 3.8.1 CPU Overview
- 3.8.2 CPU Pipeline stages
- 3.8.3 CPU Cache Organization

3.9 Branch Prediction & Folding (Concept)
3.10 GPU Overview

Unit 3: Programming using Python **10**

3.1 Overview of Rasberian OS (Operating System)
- 3.1.1 Installation
- 3.1.2 Different types of Operating Systems

3.2 Python Programming (Script programming)
- 3.2.1 Variable & data types
- 3.2.2 Flow Control structures
- 3.2.3 Conditional statements (If…Then…else)
- 3.2.4 Functions: I/O function (GPIO, Digital)
- 3.2.5 Time functions
- 3.2.6 Library functions
- 3.2.7 Basic Arithmetic Programs - Addition, Subtraction, Multiplication, Division

Unit 4 : Interfacing of devices using Python Programming **10**

4.7 Basic interfacing - LED, Switch, LCD
4.8 Internal Advanced - Bluetooth, Wifi, Ethernet
4.9 External advanced - Camera, Serial Communication GSM, Ultrasonic Sensor, PIR, Finger Print reader

References :

1. Rasberry Pi CookBook: Software & Hardware problems and Solutions By Simon Monk(O'Reilly Media Inc.)
2. Raspberry Pi Hardware Reference by Warren Gay (Apress)
3. Rasberry Pi User Guide By Eben Upton, Greath Halfacree (John Wiley & Sons, Inc.)
4. Learning Python with Rasberry Pi, by Alex Bradbury, Ben Everard, John Wiley & Sons, Inc
5. Learn Raspberry Pi programming with Python By Wolfram Donat (Apress)

Course Code and Title: USCSEL-242 Wireless Communication and Internet of Things

No. of lectures 36 (Credits 2)

Objectives:

1. To learn and understand applications of wireless communication system
2. To learn and understand cellular system
3. To learn and understand architecture of short-range Wireless Technologies
4. To learn and understand basics of Internet of Things
5. To study applications of IoT

Course Outcomes: Students will be able to

7. Know working of wireless technologies such as Mobile communication, GSM, GPRS
8. Become familiar with 3G and 4G Cellular Network Technologies for Data Connections.
9. Understand working principles of short-range communication application
10. Get introduced to upcoming technology of Internet of Things
11. Explore on their own and develop new IoT based applications

Unit1: Wireless Communication: Cellular Telephony **12**

1.1 Overview of wireless communication - Introduction of cellular telephony system

1.2 Frequency reuse, handoff strategies, Co-channel and adjacent channel interference, block diagram of mobile handset

1.3 GSM - architecture, frame structure, mobility management,

1.4 GPRS - architecture, application

Unit 2: Short Range Wireless Technologies and Location Tracking **12**

2.1 Short range Technologies:

- **2.1.1** Bluetooth
- **2.1.2** Bluetooth architecture
- **2.1.3** Bluetooth protocol stack
- **2.1.4** Bluetooth frame structure
- **2.1.5** Zigbee - Architecture, topologies, applications
- **2.1.6** Z wave - Protocol architecture, applications
- **2.1.7** RFID - working of RFID system, types of RFID tags, RFID frequencies, applications

2.2 Location Tracking

- **2.2.1** GPS system - components of GPS system (space segment, control segment, user segment), GPS receiver, Applications

Unit 3: IoT Architecture **08**

3.1 Introduction to IOT - Evolution of IOT, M2M and/or IOT,
3.2 Seven layer architecture of IoT, Role of cloud in IoT, cloud topologies, Cloud access
3.3 Protocols in IoT
3.4 Cross connectivity across IoT system components
3.4.1 Device to Gateway-short range
3.4.2 Wireless - cellphone as gateway, dedicated wireless Access points
3.4.3 Gateway to cloud - Long range connectivity, (wired, cellular, Satellite, WAN)
3.4.4 Direct Device to Cloud connectivity
3.5 Networking technologies
3.5.1 Low power local area networking (LPLAN)
3.5.2 Low power wide area networking (LPWAN) technologies.

Unit 4: IoT Applications **04**

4.10 Application domains and Challenges in IoT
4.10.1 Power consumption
4.10.2 Physical security
4.10.3 Durability
4.10.4 Secure Connectivity
4.10.5 Secure Data Storage
4.10.6 Data volume
4.10.7 Scalability
4.11 Case studies:
4.11.1 Case Study 1: Smart Irrigation system for Agricultural field
4.11.2 Case Study 2:Home Automation
4.11.3 Case Study 3: Smart Cities

References :

1. Wireless Communications Principles and Practice, Rappaport, Pearson publication
2. Mobile Communications, Jochen Schiller, Pearson publication
3. Internet of Things : Principles and Paradigms, Rajkumar Buyya and Dastjerdi, MK publishers

Course Code and Title: USCSELP-243 Electronics Laboratory
No. of sessions 12 (Credits 2)

Objectives:

1. To use basic concepts for building various applications of embedded electronics.
2. To build experimental setup and test the circuits.
3. To develop skills of analyzing test results of given experiments.
4. Developing Trained Personals for educating and training for upcoming graduates in wireless communication.
5. Implement basic IoT applications on embedded platform

Course Outcomes: On completion of the course, students will be able

1. To design and develop own smart applications using Raspberry -Pi
2. To write Python program for simple applications
3. To build own IoT based system

Guidelines :

- Practical batch size : 12
- Minimum no of Practical to be performed : 10
- Eight compulsory experiments: At least four practical from each Group
- One activity equivalent to 2 experiments by the student.
  a. Continuation of F.Y. activity.
  b. Electronics project Based on the Theory Courses learnt
  c. Documentation type experiments

Prerequisite: Raspberry Pi boards, Arduino / LoRa boards

Group A

1. Programming of Raspberry Pi to control LEDs attached to the GPIO pins
2. Programming of Raspberry Pi to get feedback from a switch connected to the GPIO pins and making on or off device.
3. Programming of Raspberry Pi to detect temperature using temperature sensor and control it at set temperature.
4. Programming of Raspberry Pi to detect light intensity using photocell sensor and control on or off of switch.

Group B

1. Study of GSM system (Message transmission & Reception) and working of SIM card in GSM handset
2. Study of Zig-bee for transmission of temperature sensed by temperature sensor
3. Study of RFID system
4. To study Arduino based LED switching using mobile and Temperature and humidity sensing using Arduino