



**Maharashtra Education Society's**  
**ABASAHEB GARWARE COLLEGE**  
**(AUTONOMOUS)**

*(Affiliated to Savitribai Phule Pune University)*

**Three Years Degree Program in Computer Application**  
**(Faculty of Science & Technology)**

**Syllabi under Autonomy**  
**S.Y.B.C.A. (Science)**

**Choice Based Credit System Syllabus**  
**To be implemented from academic year 2023-24**

**Structure of the Course: B.C.A. (Science)**

Year	Semester	Course Type	Course Code	Course Title	Remark	Credit	No. of Lectures /Practical to be conducted
2	III	CC	USCA-231	Data Structures using C		4	48
		CC	USCA-232	Python Programming		4	48
		CC	USCA-233	Computer Networks		4	48
		CC	USCAP-234	Practical Course in Data Structures using C		2	12
		CC	USCAP-235	Practical Course in Python Programming		2	12
		CC	USCAP-236	Practical Course in Computer Networks and Web Programming		2	12
		AECC	UEVS-231	Environmental Science – I		2	30
		AECC	USLGA-231	Language Communication – I		2	30
	IV	CC	USCA-241	Object Oriented Programming – I		4	48
		CC	USCA-242	Web Technology		4	48
		CC	USCA-243	Software Engineering		4	48
		CC	USCAP-244	Practical Course in Object Oriented Programming – I		2	12
		CC	USCAP-245	Practical Course in Web Technology		2	12
		CC	USCAP-246	Practical Course in Software Engineering		2	12
		AECC	UEVS-241	Environmental Science – II		2	30
		AECC	USLGA-241	Language Communication – II		2	30

**S Y B C A - Semester I****Course Code and Title: USCA-231 Data Structures using C****No. of Hours: 48 (Credits - 4)****Prerequisites:**

- Knowledge of C Programming

**Course Objectives:**

1. To understand analysis of algorithms.
2. To learn different searching and sorting techniques.
3. To understand different types of linked list.
4. To learn use of stack and queue.
5. To understand the use of tree as a data structure.
6. To learn graph and its traversal methods.

**Learning Outcomes:** On completion of the course student 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 and queue.
5. Illustrate tree terminology and its traversal techniques.
6. Explain applications of graph.

**Unit 1: Introduction to data structure****02**

- 1.1 Need of Data Structure
- 1.2 Data object, Data Structure, Abstract Data Type (ADT)
- 1.3 Types of Data Structures
- 1.4 Algorithm Analysis – Frequency counts, Space and Time complexity
- 1.5 Asymptotic notations - BigO, Omega ( $\Omega$ ), Theta ( $\theta$ )
- 1.6 Algorithms and its complexity-examples

**Unit 2: Array****10**

- 2.1 Introduction
- 2.2 Array representation - Row major and column major
- 2.3 Comparison based sorting methods - Bubble Sort, Insertion Sort, Selection Sort
- 2.4 Non-Comparison based sorting algorithms - Counting Sort, Radix Sort
- 2.5 Divide and Conquer strategy - Merge Sort, Quick Sort
- 2.6 Searching techniques with time Complexity - Linear search, Binary search

**Unit 3: Linked List** **10**

- 3.1 Introduction
- 3.2 Dynamic implementation of Linked List
- 3.3 Types of linked lists - singly, doubly, circular
- 3.4 Operations on Linked List - create, traverse, insert, delete, search, sort, reverse, concatenate, merge, time complexity of operations
- 3.5 Applications of Linked List – Polynomial representation, Addition of two polynomials

**Unit 4: Stack and Queue** **10**

- 4.1 Representation of Stack - Static and Dynamic
- 4.2 Operations on Stack – init(), push(), pop(), isEmpty(), isFull(), peek(), time complexity of all the operations
- 4.3 Applications of Stack
  - 4.3.1 Function call and recursion
  - 4.3.2 String reversal, palindrome checking, balancing of parenthesis
  - 4.3.3 Expression types - infix, prefix and postfix, expression conversion and evaluation (implementation of infix to postfix and evaluation of postfix expression)
- 4.4 Representation of Queues - Static and Dynamic
- 4.5 Operations on queue - insert, delete, empty, full, peek
- 4.6 Types of Queue - Linear Queue, Circular Queue, Priority Queue, Double Ended Queue
- 4.7 Application of queue – Priority Queue, CPU scheduling

**Unit 5: Tree** **10**

- 5.1 Introduction and Tree terminologies
- 5.2 Binary trees: Types - full, complete and skewed
- 5.3 Representation of Binary Trees - Static and Dynamic
- 5.4 Types of Traversal - Preorder, Inorder, Postorder (Recursive implementation)
- 5.5 Application of Binary trees – AVL tree
- 5.6 Binary Search Tree (BST) and operations – create, insert, search, mirror

**Unit 6: Graph** **6**

- 6.1 Introduction and Graph terminologies
- 6.2 Representation of a Graph – Adjacency matrix, Adjacency list, Adjacency multi-list
- 6.3 Graph Traversals – Breadth First Search and Depth First Search
- 6.4 Application of Graph – Topological sort, Dijkstra's Algorithm

**Reference Books**

1. Fundamentals of Data Structures - Horowitz Sahani (Galgotia)
2. Introduction to Data Structures using C - Ashok Kamthane

3. Data Structures using C - Bandopadhyay & Dey(Pearson)
4. Data Structures using C - Srivastava BPB Publication

**Course Code and Title: USCA-232 Python Programming****No. of Hours: 48 (Credits - 4)****Prerequisites:** None**Course Objectives:**

1. To introduce programming concepts using python
2. Student should be able to develop Programming logic using python
3. To develop basic concepts and terminology of python programming
4. To test and execute python programs

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

1. Develop logic for problem solving
2. Determine the methods to create and develop Python programs by utilizing the data structures like lists, dictionaries, tuples and sets.
3. Become familiar about the basic constructs of programming such as data, operations, conditions, loops, functions etc.
4. Write python programs and develop a small application project.

**Unit 1: Introduction to Python****04**

- 1.1 Features, Applications
- 1.2 Installation and Working with Python
- 1.3 Python IDEs
- 1.4 Numeric data type – int, float, complex
- 1.5 Strings - declaration, manipulation, special operations, escape character, string formatting operator, Raw String, Unicode strings, Built-in String methods
- 1.6 Basic operators, Membership operators, Identity operators, Expressions, comments
- 1.7 Type conversion

**Unit 2: Data structures and Control statements****12**

- 2.1 List - Concept, creating list, accessing, updating & deleting an element from list, traversing a List, Built-in List functions and methods, List comprehension
- 2.2 Tuple - Creating tuple, Accessing values in Tuple, Tuple Assignment, Basic tuple operations, Built-in tuple functions
- 2.3 Dictionary - Creating a Dictionary, Accessing Values in a dictionary, Updating Dictionary, Deleting Elements from Dictionary, Properties of Dictionary keys, Built-in Dictionary Methods
- 2.4 Set – Creating set, accessing element, operations of Set (union, intersection, difference, symmetric-difference), working with sets
- 2.5 Conditional Statements: if, if-else, elif
- 2.6 Looping- for, while, nested loops
- 2.7 Loop control statements (break, continue)

## 2.8 pass statement

**Unit 3: Function****06**

- 3.1 Concept, Function Calls, Flow of Execution, Arguments and its types, function composition, Anonymous functions
- 3.2 Importing with from, Return Values, Boolean Functions, Recursion, Functional programming tools - filter(), map(), and reduce(), recursion, lambda forms

**Unit 4: Modules and Packages****06**

- 4.1 Modules: Importing module, Creating & exploring modules, Math module, Random module, Time module
- 4.2 Packages: Importing package, creating package, examples
- 4.3 Numpy - Introduction to numPy
- 4.4 Creating NumPy Array
- 4.5 NumPy Array Manipulation
- 4.6 Matrix in NumPy
- 4.7 Operations on NumPy Array
- 4.8 Reshaping of Array

**Unit 5: Files and Exception Handling****10**

- 5.1 Working with files: Creating files and Operations on files (open, close, read, write), File object attributes, file positions, Listing Files in a Directory, Testing File Types, Removing files and directories, copying and renaming files, splitting pathnames, creating and moving directories
- 5.2 Regular Expression- Concept of regular expression, various types of regular expressions, using match function
- 5.3 Exception Handling: Built-in Exceptions, Handling Exceptions, Exception with Arguments, User-defined Exceptions

**Unit 6: Database Connectivity****04**

- 6.1 SQL Database connection using python
- 6.2 Creating and searching tables
- 6.3 Reading and storing config information on database
- 6.4 Programming using database connections

**Unit 7: GUI Programming****06**

- 7.1 Introduction
- 7.2 Tkinter Programming
- 7.3 Tkinter Widgets
- 7.4 Frame, Button, Label, Entry, Canvas, Check button, listbox, menu, text

**Reference Books:**

1. An Introduction to Computer Science using Python 3 by Jason Montojo, JenniferCampbell, Paul Gries, The pragmatic bookshelf-2013
2. James Payne, “Beginning Python”: Using Python and Python 3.1,Wrox Publication
3. Introduction to Computer Science Using Python- Charles Dierbach, Wiley PublicationLearning with Python “, Green Tea Press, 2002
4. Introduction to Problem Solving with Python by E balguruswamy, TMH publication-2016
5. Beginning Programming with Python for Dummies Paperback – 2015 by John PaulMueller
6. Object-oriented Programming in Python, Michael H. Goldwasser, David Letscher, Pearson Prentice Hall-2008
7. Numpy - <https://ncert.nic.in/textbook/pdf/keip106.pdf>

**Course Code and Title: USCA-233 Computer Networks****No. of Hours: 48 (Credits-04)****Prerequisite Courses:** None**Course Objectives:**

1. To provide the fundamental concepts of networking standards and technology.
2. To gain the knowledge of OSI/IP, TCP/IP model and addressing techniques.
3. To learning various line encoding schemes and switching methods
4. To study different techniques for framing, error control, flow control and routing.
5. To learning the network protocols.
6. To develop an understanding of modern network architectures from a design and performance perspective.

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

1. Analyze the requirements for a given organization and select appropriate network architecture, topologies, transmission mediums and technologies.
2. Understand different network model and addressing techniques.
3. Understand various line encoding schemes and switching methods
4. Understand different techniques for framing, error control, flow control and routing.
5. Understand the IPV4 and IPV6 network protocols.
6. Illustrate applications of Computer Network.

**Unit 1: Introduction to Data Communication****08**

- 1.1 Concept, Characteristics and Components of Data communication
- 1.2 Types of Data flow – Simplex, Half Duplex, Full Duplex
- 1.3 Computer Networks applications –Business Application, Home Application, Mobiles
- 1.4 Broadcast and point-to-point networks
- 1.5 Network Topologies - Bus, Star, Ring, Mesh
- 1.6 Network Types- LAN, MAN, WAN, PAN, Wireless Networks, Home Networks, internetworks

**Unit 2: Network Models****04**

- 2.1 OSI Model – layered architecture, peer-to-peer processes, encapsulation
- 2.2 TCP/IP Model – layers and Protocol Suite
- 2.3 Addressing-Physical, Logical, Port addresses, Specific addresses

**Unit 3: Physical Layer****10**

- 3.1 Analog and Digital data, Analog and Digital signals, Digital Signals-Bit rate, Bit length
- 3.2 Baseband Transmission, Broadband Transmission
- 3.3 Transmission Impairments– Attenuation, Distortion and Noise
- 3.4 Data Rate Limits– Noiseless channel: Nyquist's bit rate, noisy channel: Shannon's law
- 3.5 Performance of the Network Bandwidth, Throughput, Latency (Delay), Bandwidth – Delay Product, Jitters
- 3.6 Line Coding Characteristics, Line Coding Schemes–Unipolar -NRZ, Polar-NRZ-I, NRZ-L, RZ, Manchester and Differential Manchester, Problems on coding schemes
- 3.7 Transmission Modes, Parallel Transmission and Serial Transmission– Asynchronous, Synchronous and Isochronous
- 3.8 Switching-Circuit Switching, Message Switching and Packet Switching.

**Unit 4: Data Link Layer****06**

- 4.1 Framing – Concept, Methods – Character Count, Flag bytes with Byte Stuffing, Starting and ending Flags with Bit Stuffing
- 4.2 Elementary data link protocols - Simplex stop and wait protocol, Simplex protocol for noisy channel
- 4.3 Sliding Window Protocols – 1-bit sliding, Pipelining Go-Back N and Selective Repeat
- 4.4 Random Access Protocols - p-persistent and non-persistent CSMA/CD, CSMA/CA
- 4.5 Controlled Access - Reservation, Polling and Token Passing

**Unit 5: Network Layer****08**

- 5.1 IPv4 addresses: Address space, Notation, Classful addressing, Classless addressing, NAT, Sub netting, Super netting
- 5.2 IPv4: Datagram, Fragmentation, checksum, options
- 5.3 IPv6 addresses: Structure, address space
- 5.4 IPv6: packet format, Extension headers

**Unit 6: Transport and Application Layer****12**

- 6.1 User Datagram Protocol (UDP) - Datagram Format, Checksum, UDP operations, Use of UDP
- 6.2 Transmission Control Protocol (TCP) - TCP Services – Process to-Process Communication, Stream Delivery Service- Sending and Receiving Buffers and Segments
- 6.3 TCP Features – Numbering System, Byte Number, Sequence Number, Acknowledgement Number, Flow Control, Error Control, Congestion Control
- 6.4 TCP Segment Format
- 6.5 TCP Vs UDP
- 6.6 Domain Name System (DNS) - Distribution of Name Space, DNS in the Internet
- 6.7 E-MAIL - Architecture, User Agent, Message Transfer Agent – SMTP
- 6.8 FTP

## 6.9 HTTP- HTTP Transaction

### **Reference Books:**

1. Data Communications and Networking by Behrouz Forouzan, Fifth Edition, ISBN 978-0-07-337622-6 McGraw Hill.
2. Computer Networks, ANDREW S. Tanenbaum, Fifth Edition, ISBN-13: 978-0-13-212695-3, Pearson

**Course Code and Title: USCAP-234 Practical course in Data Structures using C**

**No. of Sessions: 12 (Credits-2)**

**Assignments:**

1. Non recursive sorting techniques: Bubble sort, Insertion sort, Count sort
2. Recursive Sorting Techniques: Merge sort, Quick sort
3. Searching techniques: Linear search, Binary search
4. Link List: Singly, Singly circular, Doubly, Doubly circular with operations – Create, insert, delete, search, Polynomial representation and operations
5. Stack: Static and dynamic implementation
6. Queue: Static and dynamic implementation
7. Binary Search Tree(BST) – Dynamic implementation, Operations - Create, insert, search, count leaf, non leaf and total nodes, and mirror
8. Graph: Adjacency Matrix Representation, Adjacency List Representation, In-degree and Out-degree calculation

**Course Code and Title: USCAP-235 Practical Course in Python  
Programming**

**No. of Sessions: 12 (Credits-2)**

**Assignments:**

1. Python Basics
2. String
3. List, Tuples, Sets, and Dictionary
4. Functions
5. Numpy – Array manipulation, Matrix, Operation on Numpy array
6. File Handling
7. Exception handling
8. Regular expression
9. Database handling
10. Using GUI

**Course Code and Title: USCAP-236 Practical course in Web  
Technology**

**No. of Hours: 24 (Credits - 2)**

**Prerequisites:**

- Knowledge of HTML, CSS, JavaScript

**Course Objectives:**

1. To understand components of World Wide Web as a platform.
2. To understand the basics of PHP.
3. To study library and user defined functions.
4. To learn the use of array in PHP.

**Learning Outcomes:** On completion of the course student will be able to

1. Distinguish between server-side and client-side web technologies.
2. Know the basic structure of PHP.
3. Apply appropriate function for a real-world problem.
4. Apply the use of arrays for different applications.

**Unit 1: Introduction to Web Technologies****4**

- 1.1 Qualities of a good website
- 1.2 Software to create websites
- 1.3 Web Client-Server and its Communication
- 1.4 Client and Server Scripting Languages
- 1.5 HTTP request message, HTTP response message
- 1.6 Types of Websites (Static and Dynamic Websites)

**Unit 2: Introduction to PHP****4**

- 2.1 Introduction to PHP
- 2.2 What does PHP do?
- 2.3 Lexical structure
- 2.4 Language basics

**Unit 3: Function and String****10**

- 3.1 Defining and calling a function
- 3.2 Default parameters
- 3.3 Variable parameters, Missing parameters
- 3.4 Variable function, Anonymous function
- 3.5 Types of strings in PHP
- 3.6 Printing functions
- 3.7 Encoding and escaping

- 3.8 Comparing strings
- 3.9 Manipulating and searching strings
- 3.10 Regular expressions

**Unit 4: Arrays****6**

- 4.1 Indexed Vs Associative arrays
- 4.2 Identifying elements of an array
- 4.3 Storing data in arrays
- 4.4 Multidimensional arrays
- 4.5 Extracting multiple values
- 4.6 Converting between arrays and variables
- 4.7 Traversing arrays
- 4.8 Sorting
- 4.9 Action on entire arrays
- 4.10 Use of arrays – Set, Stack and Queue

**Reference Books**

- 1. Programming PHP By Rasmus Lerdorf and Kevin Tatroe, O'Reilly publication
- 2. Beginning PHP5, Wrox publication
- 3. PHP for Beginners, SPD publication
- 4. Mastering PHP BPB Publication

Unit 2: Book 1-Chapter 2

Unit 3: Book 1-Chapter 3 and 4

Unit 4: Book 2-Chapter 5

**Practical Component****No. of Sessions: 12****Assignments:**

1. Using Networking commands – ping, hostname, traceroute, netstat, ifconfig, who, nmap, nslookup IP, MAC addresses
2. Basic assignments on PHP
3. Function in PHP
4. String in PHP
5. Arrays in PHP

**Note:**

- Theory lectures will be conducted for the whole class at a time.
- Lab Sessions will be conducted for each batch of 12 students separately.
- Practical exam of 35 marks will be conducted based on assignments carried out during lab sessions only.
- Continuous assessment of 15 marks will be carried out based on performance in both lab and theory sessions.

**S Y B C A - Semester II****Course Code and Title: USCA- 241 Object Oriented programming I****No. of Hours: 48 (Credits-4)****Prerequisites:** None**Course Objectives:**

- 1 To introduce the object-oriented programming concepts.
- 2 To understand object-oriented programming concepts, and apply them in solving problems.
- 3 To introduce the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes
- 4 To introduce the implementation of packages and interfaces
- 5 To introduce the concepts of exception handling and multithreading.
- 6 To introduce the design of Graphical User Interface using swing controls

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

1. Understand the concept of classes, object, packages and file handling.
2. Develop GUI based applications using Java.

**Unit 1: Introduction to Object oriented programming (OOP)****08**

- 1.1 Object Oriented Programming Concepts
- 1.2 Features of OOP
- 1.3 Application and advantages of OOP
- 1.4 Introduction of JAVA: history of Java, Features OR Buzzwords of Java, Java Environment
- 1.5 Types of Comments
- 1.6 Java Tools – jdb, javap, javadoc
- 1.7 Data Types
- 1.8 Final Variable
- 1.9 Declaring 1D, 2D Array
- 1.10 Accepting Input (Command Line Arguments, BufferedReader, Scanner)

**Unit 2: Classes and Objects****12**

- 2.1 Defining a class
- 2.2 Access Specifiers (public, protected, private, default)
- 2.3 Array of Objects
- 2.4 Constructors, Overloading Constructors and Use of 'this' keyword
- 2.5 Concept of Garbage Collection
- 2.6 static block, static fields and methods
- 2.7 Predefined Classes
  - 2.7.1 Object Class, Methods (equals(), toString(), hashCode(), getClass())
  - 2.7.2 String Class and StringBuffer Class, Formatting String data using format() method

2.8 Creating, Accessing and Using Packages

2.9 Wrapper Classes

### **Unit 3: Inheritance and Interface**

**12**

3.1 Inheritance Basics (extends Keyword) and Types of Inheritance

3.2 Superclass, Subclass and use of Super Keyword

3.3 Method Overriding and runtime polymorphism

3.4 Use of final keyword related to method and class

3.5 Use of abstract class and abstract methods

3.6 Defining and Implementing Interfaces

3.7 Runtime polymorphism using interface

3.8 Concept of Marker and Functional Interfaces

### **Unit 4: Exception and File Handling**

**08**

4.1 Dealing with errors , Exception class, Checked And Unchecked Exception

4.2 Catching Exceptions, Multiple Catch Block, Nested try block

4.3 Creating User Defined Exception

4.4 Introduction to Files and Streams

Input-OutputStream : FileInputStream/OutputStream, BufferedInput/OutputStream,  
DataInput/OutputStream

4.5 Reader-Writer : FileReader/Writer, BufferedReader/Writer, InputStreamReader,  
OutputStreamWriter

### **Unit 5: AWT and Swing**

**08**

5.1 Concept of AWT and Swing

5.2 Difference between AWT and Swing

5.3 The MVC Architecture and Swing

5.4 Layouts and Layout Managers

5.5 Containers and Components – JFrame, JButton, JLabel, JText, JTextArea, JCheckBox  
and JRadioButton, JList, JComboBox, JMenu And related Classes

5.6 Dialogs (Message, Confirmation, Input), JFileChooser, JColorChooser

5.7 Event Handling: Event Sources, Listeners

5.8 Adapters And Anonymous Inner Class

### **Reference Books:**

1 Complete reference Java by Herbert Schildt (5th edition)

2 Java 2 programming black books, Steven Horlner

3 Programming with Java , A primer ,Forth edition , By E. Balagurusamy

4 Core Java Volume-I-Fundamentals, Eighth Edition, Cay S. Horstmann, Gary Cornell,  
Prentice Hall, Sun Microsystems Press

**Course Code and Title: USCA-242 Advanced Web Technology**  
**No. of Hours: 48 (Credits - 4)**

**Prerequisites:**

- Knowledge of HTML, CSS, JavaScript and Basics of PHP

**Course Objectives:**

1. To Design dynamic and interactive Web pages.
2. To learn different ways dealing with Files and Directories in PHP.
3. To learn database connectivity and interaction of PHP program with database.
4. To understand the use of JSON object.
5. To understand and learn AJAX.

**Learning Outcomes:** On completion of the course student will be able to

1. Develop web based applications using suitable client side and server side web technologies by maintaining state of the user.
2. Store and extract the appropriate information from files and directories.
3. Build Dynamic web site using server-side PHP Programming and Database connectivity.
4. Create appropriate JSON object for sending request and response.
5. Build applications using AJAX as a single page website.

**Unit 1: Web Techniques** **8**

- 1.1 Variables
- 1.2 Server information
- 1.3 Processing forms
- 1.4 Setting response headers
- 1.5 Maintaining state

**Unit 2: Files and directories** **8**

- 2.1 Working with files and directories
- 2.2 Opening and Closing, Getting information about file, Read/write to file,
- 2.3 Splitting name and path from file, Rename and delete files
- 2.4 Reading and writing characters in file
- 2.5 Reading entire file
- 2.6 Random access to file data
- 2.7 Getting information on file
- 2.8 Ownership and permissions

**Unit 3: Databases** **8**

- 5.1 Using PHP to access a database

5.2 Relational databases and SQL

5.3 PEAR DB basics

#### **Unit 4: PHP and JSON**

**6**

4.1 Introduction to JSON,

4.2 JSON Data types

4.3 JSON Object

4.4 JSON schema

4.5 JSON Comparison with XML

4.6 Encoding and Decoding JSON in PHP

#### **Unit 5: Ajax**

**8**

5.1 Introduction of AJAX

5.2 AJAX web application model

5.3 Performing AJAX validation

5.4 Handling JSON data using php and AJAX

5.5 Connecting database using php and AJAX

#### **Unit 6: PHP framework CodeIgniter**

**10**

6.1 CodeIgniter – Overview

6.2 Installing CodeIgniter

6.3 Application Architecture

6.4 MVC Framework

6.5 Basic concept of CodeIgniter

6.6 Libraries Working with databases

6.7 Load external JS and CSS page & redirecting from controller

6.8 Adding JS and CSS, Page redirection

6.9 Loading dynamic data on page & session management, cookies management

#### **Reference Books**

1. Programming PHP By Rasmus Lerdorf and Kevin Tatroe, O'Reilly publication
2. Beginning PHP5, Wrox publication
3. PHP for Beginners, SPD publication
4. AJAX Black Book Kogent solution
5. Professional Codeigniter By Thomas Myer ,Wrox Publication
6. Codeigniter 2 CookBook By Rob Foster, PACKT Publication

Unit 1: Book 1-Chapter 7

Unit 2: Book 2-Chapter 7

Unit 3: Book 1-Chapter 9

Unit 4: Book 2-Chapter 8

Unit 5: Book 4 Chapter 1,2 and 9

**Course Code and Title: USCA-243 Software Engineering**  
**Lectures: 48 (Credits-4)**

**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** **05**

- 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** **05**

- 2.1 Concept of Agile software development

- 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****08**

- 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****10**

- 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 Modeling****08**

- 5.1 Use case diagram
- 5.2 Activity diagram
- 5.3 State diagram
- 5.4 Sequence diagram.

<b>Unit 6: Architectural Modeling</b>	<b>08</b>
6.1 Component Diagram	
6.2 Deployment Diagram	
<b>Unit 7: Case Studies</b>	<b>04</b>

**Reference Books:**

1. Software Engineering: A Practitioner's Approach- Roger S. Pressman, McGraw hill International Editions 2010(Eighth Edition)
2. Grady Booch, James Rumbaugh, 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: USCAP- 244 Practical Course in Object Oriented programming I**

**No. of Sessions : 12 (Credits-2)**

**Assignments :**

1. Java Tools and IDE, Simple Java Programs
2. Array of Objects and Packages
3. Inheritance and Interfaces
4. Exception and File Handling
5. GUI Designing, Event Handling

**Course Code and Title: USCAP-245 Practical course in  
Advanced Web Technology**

**No. of Sessions: 12 (Credits - 2)**

**Assignments:**

1. Self-Processing, Sticky forms and File upload
2. COOKIES and SESSIONS
3. File Handling
4. XML documents and DOM
5. AJAX
6. PHP framework CodeIgniter

**Course Code and Title: USCAP-246 Practical Course in Software Engineering**  
**No. of Sessions – 12 (Credits – 2)**

**Assignments of Mini Project using Software Engineering Techniques:**

1. Structural Model :
  - Problem definition, scope of proposed system
  - Requirement Specification
  - ER Diagram
  - Use Case Diagram
  - Class Diagrams
2. Behavioral Model :
  - Sequence Diagram
  - Collaboration Diagram
  - Activity Diagram
  - State Chart Diagram
3. Architectural Model :
  - Component Diagram
  - Deployment Diagram
  - Package Diagram