Program: B.Sc. - Computer Science					**Semester : IV**
Course: Fundamentals of Algorithms					**Course Code: USMACS401**
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1 = 3	25	75

Learning Objectives:

- To understand basic principles of algorithm design and why algorithm analysis is important
- To understand how to transform new problems into algorithmic problems with efficient solutions
- To understand algorithm design techniques for solving different problems

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Examine and evaluate performance of different algorithms

CO2: Derive and solve recurrences describing the performance of divide-and-conquer algorithms

CO3: Analyze and apply string matching algorithm.

CO4: Design optimal solution by applying various algorithm techniques like Dynamic Programming and Greedy Method.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	Algorithm Analysis	10
2	Tree, String Matching and Selection Algorithms	10
3	Algorithms Design Techniques	10
	Total	**30**
PRACTICALS		**30**

Module	Fundamental of Algorithms	No. of Hours/Credits 30/2
1	**Algorithm Analysis**	**10**
	Introduction to algorithm, Why to analysis algorithm, Running time analysis, How to Compare Algorithms, Rate of Growth, Commonly Used Rates of Growth, Types of Analysis, Asymptotic Notation, Big-O Notation, Omega-Ω Notation, Theta-Θ Notation, Asymptotic Analysis, Properties of Notations, Commonly used Logarithms and Summations, Performance characteristics of algorithms, Master Theorem for Divide and Conquer, Divide and Conquer Master Theorem: Problems & Solutions, Master Theorem for Subtract and Conquer Recurrences	
2	**Tree, String Matching and Selection Algorithms**	**10**
	Generic Trees (N-ary Trees), Threaded Binary Tree, Binary Search Trees (BSTs), Balanced Binary Search Trees, AVL (Adelson-Velskii and Landis) Trees, Heapsort String Matching: Introduction, The naive string-matching algorithm, The Rabin-Karp algorithm, String Matching with finite automata, The Knuth-Morris-Pratt algorithm Selection Algorithms: What are Selection Algorithms? Selection by Sorting, Partition-based Selection Algorithm, Linear Selection Algorithm - Median of Medians Algorithm, Finding the K Smallest Elements in Sorted Order	
3	**Algorithms Design Techniques**	**10**
	Algorithms Design Techniques: Introduction, Classification, Classification by Implementation Method, Classification by Design Method Greedy Algorithms: Introduction, Greedy Strategy, Elements of Greedy Algorithms, Advantages and Disadvantages of Greedy Method, Greedy Applications, Understanding Greedy Technique Divide and Conquer Algorithms: Introduction, What is Divide and Conquer Strategy? Divide and Conquer Visualization, Understanding Divide and Conquer, Advantages of Divide and Conquer, Disadvantages of Divide and Conquer, Divide and Conquer Applications Dynamic Programming: Introduction, what is Dynamic Programming Strategy? Properties of Dynamic Programming Strategy, Problems which can be solved using Dynamic Programming - Longest Common Subsequence, Dynamic Programming Approaches	

PRACTICALS

Sr. No.	Topic.
1	Write Python program to sort n names using Quick sort algorithm. Discuss the complexity of algorithm used.
2	Write Python program to sort n numbers using Merge sort algorithm. Discuss the complexity of algorithm used.
3	Write a Python program to implement Heapsort.
4	Write a python program to implement Rabin-Karp algorithm.
5	Write a python program to implement KMP algorithm.
6	Write Python program for finding the second largest element in an array A of size n using Tournament Method.
7	Write python program to find kth smallest element using partition-based algorithm.
8	Write Python program for implementing Huffman Coding Algorithm. Discuss the complexity of algorithm.

RECOMMENDED READING:

Text Books:

1. Data Structure and Algorithmic Thinking with Python, Narasimha Karumanchi , CareerMonk Publications, 2016
2. Introduction to Algorithm, Thomas H Cormen, PHI
3. Design and analysis of algorithms, Himanshu Dave, Pearson, 2nd Edition

Reference Books

1. Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, 2016, Wiley
2. Fundamentals of Computer Algorithms, Sartaj Sahni and Sanguthevar Rajasekaran Ellis Horowitz, Universities Press
3. Introduction to Design and Analysis of Algorithms, Anany Levitin, Pearson

Program: B.Sc.- Computer Science	**Semester : IV**
Course: Advanced Java	**Course Code: USMACS402**

Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1	25	75

Learning Objectives:

- To Explore swing API for building GUI applications.
- To Understand the Java Database Connectivity.
- To Understand Web application development using Servlet, JSP and sturts2.

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Apply swing concepts for creating GUI java applications.

CO2: Implement database connectivity with JDBC.

CO3: Explore servlets, JSP for web development.

CO4: Recognize java beans, struts2 framework and JSON format.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	GUI using Swing & JDBC	10
2	Web Application development with Servlet & JSP	10
3	Java Sturts2 framework & JSON	10
	Total	**30**
PRACTICALS		**30**

Module	**Advanced Java**	**No. of Hours/Credits 30/2**
1	**GUI using Swing & JDBC**	**10**
	Swing: Need for swing components, Difference between AWT and swing, Components hierarchy, Panes, Swing components: Jlabel, JTextField and JPasswordField, JTextArea, JButton, JCheckBox, JRadioButton, JComboBox and JList. **JDBC**: Introduction, JDBC Architecture, Types of Drivers, Statement, ResultSet, Read Only ResultSet, Updatable ResultSet, Forward Only ResultSet, Scrollable ResultSet, PreparedStatement, Connection Modes, SavePoint, Batch Updations, CallableStatement.	
2	**Web Application development with Servlet & JSP**	**10**
	Servlets: Introduction, Http Methods, Web Server & Web Container, Servlet Interface, GenericServlet, HttpServlet, Servlet Life Cycle, ServletConfig, ServletContext, Servlet Communication, Session Tracking Mechanisms **JSP**: Introduction, JSP LifeCycle, JSP Implicit Objects & Scopes, JSP Directives, JSP Scripting Elements, JSP Actions: Standard actions and customized actions.	
3	**Java Beans, Sturts2 framework & JSON**	**10**
	Java Beans:-Introduction to JavaBeans-Properties. **Struts 2**: Basic MVC Architecture, Struts 2 framework features, Struts 2 MVC pattern, Request life cycle, Examples, Configuration Files, Actions, Interceptors, Results & Result Types, Value Stack/OGNL. **JSON**: Overview, Syntax, Data Types, Objects, Schema, Comparison with XML, JSON with Java.	

PRACTICALS	
Sr. No.	**Topic.**
1	Develop the presentation layer of Library Management software application with suitable menus.
2	Develop business logic layer for Library Management System:-1.
3	Develop business logic layer for Library Management System:-2.
4	Write a Java application to demonstrate servlet life cycle.
5	Design a servlet session management application using all the 4 types studied.
6	Implement CRUD operations using servlet(s).
7	Implement CRUD operations using JSP(s).
8	Write a Student class with three properties. The useBean action declares a JavaBean for use in a JSP. Write Java application to access JavaBeans Properties.
9	Design application using Struts2. Application must accept user name and greet user when command button is pressed.
10	Demonstrate a MVC application using Struts2.

RECOMMENDED READING:

Textbooks:

1) Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II–Advanced Features Prentice Hall PTR,9 th Edition

2) Herbert Schildt, Java2: The Complete Reference, Tata McGraw-Hill,5 th Edition

3) Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD) ,3 rd Edition.

Reference Books:

1) Advanced Java Programming, Uttam K. Roy, Oxford University Press

2) The Java Tutorials: http://docs.oracle.com/javase/tutorial/)

3) The Java Tutorials of Sun Microsystems Inc

Program: B.Sc.- Computer Science					**Semester : IV**
Course: Physical Computing and IoT Programming					**Course Code: USMACS403**
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1=3	25	75

Learning Objectives:

To learn about SoC architectures; Learn how Raspberry Pi. Learn to program Raspberry Pi. Implementation of internet of Things and Protocols

Course Outcomes:

CO1 : Enable learners to understand System On Chip Architectures.

CO2 : Introduction and preparing Raspberry Pi with hardware and installation.

CO3: Learn physical interfaces and electronics of Raspberry Pi and program them using practical's

CO4 : Learn how to make consumer grade IoT safe and secure with proper use of protocols.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	**Introduction to Microcontroller, Raspberry Pi**	10
2	**M2M to IoT _ A Market, M2M and IoT Technology Fundamentals**	10
3	**IoT and Protocols, IoT Security and Interoperability**	10
	Total	**30**
PRACTICALS		**30**

Module	**Physical Computing and IoT Programming**	**No. of Hours/Credits 30/2**
1	**Introduction to Microcontroller, Embedded system and micro processors**	**10**
	. **Characteristics microcontroller.** **SoC and Raspberry Pi** **System on Chip** : What is System on chip? Structure of System on Chip. **SoC products** : FPGA, GPU, APU, Compute Units. **ARM 8 Architecture** : SoC on ARM 8. ARM 8 Architecture Introduction **Introduction to Raspberry Pi** : Introduction to Raspberry Pi, Raspberry Pi Hardware, Preparing your raspberry Pi. **Programming Raspberry Pi** **Raspberry Pi and Linux:** About Raspbian, Configuring Raspberry Pi with Linux Commands **Programing interfaces:** Introduction to Node.js, **Raspberry Pi Interfaces:** UART, GPIO, I2C, SPI **Useful Implementations:** Cross Compilation, Pulse Width Modulation, SPI for Camera.	
2	**M2M to IoT _ A Market, M2M and IoT Technology Fundamentals**	**10**
	Introduction of M to M : A brief background M2M communication A typical M2M solution overview Key application areas Trends in information and communications technologies **M2M to IoT _ A Market** Perspective Information marketplaces ,Global value chains, M2M value chains, IoT value chains M2M to IoT _ An Architectural Overview Building an architecture **M2M and IoT Technology Fundamentals** Devices and gateways , Local and wide area networking, Data management, M2M and IoT analytics, Knowledge management, **Architecture Reference Model** IoT reference model, Information model, Functional model, Communication model, Safety, privacy, trust, security model	
3	**IoT and Protocols, IoT Security and Interoperability**	**10**
	Introduction to IoT: What is IoT? IoT examples, Simple IoT LED Program. **IoT and Protocols** **IoT Security:** UPnp, CoAP, MQTT, XMPP . **IoT Service as a Platform:** Clayster, Thinger.io, SenseIoT, carriots and Node RED. **IoT Security and Interoperability:** Risks, Modes of Attacks, Tools for Security and Interoperability.	

PRACTICALS	
Sr. No.	**Topic.**
1	Study of Linux Commands
2	Study of different IC's (8255 , 8259, 8237 and UART)
3	Study and Understanding of Raspberry Pi
4	GPIO: Light the LED with Python
5	Stepper Motor Control: PWM to manage stepper motor speed
6	Study of software used in raspberry pi
7	Case study on IoT in different fields
8	Study / learn any application using raspberry pi
9	GPIO : to control the brightness of LED using Raspberry Pi
10	Stack of Raspberry Pi for better Computing and analysis

RECOMMENDED READING:

Text Books:

1) Learning Internet of Things, Peter Waher, Packt Publishing(2015)
2) Mastering the Raspberry Pi, Warren Gay, Apress(2014)

Reference Books

1. Abusing the Internet of Things, Nitesh Dhanjani, O'Reilly

Program: B.Sc.- Computer Science					**Semester : IV**
Course: Android Developer Fundamentals					**Course Code: USMACS404**
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1=3	25	75

Learning Objectives:

To provide the comprehensive insight into developing applications running on smart mobile devices and demonstrate programming skills for managing task on mobile. To provide systematic approach for studying definition, methods and its applications for Mobile-App development.

Course Outcomes:

CO1 : Understand the requirements of Mobile programming environment.
CO2 : Learn about basic methods, tools and techniques for developing Apps
CO3 : Explore and practice App development on Android Platform
CO4 : Develop working prototypes of working systems for various uses in daily lives.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	**Android, Activities and Intents**	10
2	**User Input Controls**	10
3	**Data, ContentProviders**	10
	Total	**30**
PRACTICALS		**30**

Module	Android Developer Fundamentals	No. of Hours/Credits 30/2
1	**Android, Activities and Intents**	**10**
	What is Android? Obtaining the required tools, creating first android app, understanding the components of screen, adapting display orientation, action bar, Activities and Intents, Activity Lifecycle and Saving State, Basic Views: TextView, Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton, and RadioGroup Views, ProgressBar View, AutoCompleteTextView, TimePicker View, DatePicker View, ListView View, Spinner View	
2	**User Input Controls**	**10**
	User Input Controls, Menus, Screen Navigation, RecyclerView, Drawables, Themes and Styles, Material design, Providing resources for adaptive layouts, AsyncTask and AsyncTaskLoader, Connecting to the Internet, Broadcast receivers, Services, Notifications, Alarm managers, Transferring data efficiently	
3	**Data, ContentProviders**	**10**
	Data - saving, retrieving, and loading: Overview to storing data, Shared preferences, SQLite primer, store data using SQLite database, ContentProviders, loaders to load and display data, Permissions, performance and security, Firebase and AdMob, Publish your app	

PRACTICALS	
Sr. No.	**Topic.**
1	Install Android Studio and Run Hello World Program
2	Create an android app with Interactive User Interface using Layouts
3	Create an android app that demonstrates working with TextView Elements
4	Create an android app that demonstrates Activity Lifecycle and Instance State.
5	Create an android app that demonstrates the use of Keyboards, Input Controls, Alerts, and Pickers
6	Create an android app that demonstrates the use of an Options Menu
7	Create an android app that demonstrate Screen Navigation Using the App Bar and Tabs
8	Create an android app to Connect to the Internet and use BroadcastReceiver
9	Create an android app to show Notifications and Alarm manager
10	Create an android app to save user data in a database and use of different queries

RECOMMENDED READING:

Text Books:

1) "Beginning Android 4 Application Development", Wei-Meng Lee, March 2012, WROX.

Reference Books

1) https://developers.google.com/training/courses/android-fundamentals

2) https://www.gitbook.com/book/google-developer-training/android-developer-fundamentals-course-practicals/details

Program: B. Sc. - Computer Science	**Semester: IV**
Course: Linear Algebra with Python	**Course Code: USMACS405**

Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2 + 1 =3	25	75

Learning Objectives:

- To offer the learner the relevant linear algebra concepts through computer science applications.

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Students should be able to solve linear equations and do various matrix computations

CO2: Students should be able to derive Echelon form, Row canonical form, and deal with basis and change of basis computations

CO3: Students should be able to calculate eigenvalues and diagonalize using them and do various linear transformations.

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	Vectors, Linear Equations, Matrices	10
2	Solving Systems of Linear Equations, Vector spaces, Vector basis	10
3	Eigenvalues, Eigenvectors and Diagonalization, Inner Product, Orthogonality	10
	Total	**30**
PRACTICALS		**30**

Module	Linear Algebra with Python	No. of Hours/Credits 30/2
1	**Vectors, Linear Equations, Matrices**	**10**
	Vectors: Introduction, Vector addition and multiplication, Dot-product, Cross Product, The geometry of sets of vectors, Complex numbers Matrices: Matrices as vectors, Dot Product, Matrix Multiplication, Transpose, Inverse, Matrix Transformations, Determinant and its properties Linear Equations: Basics, Linear Systems of Equations, Homogeneous Systems and otherwise.	
2	**Solving Systems of Linear Equations, Vector spaces, Vector basis**	**10**
	Solving Systems of Linear Equations: Solving triangular system of linear equations, Gaussian Elimination, Echelon form, Row Canonical form Vector spaces: Vector Spaces, Subspaces, Linear Combinations, Vector Span and Spanning Set, Linear Dependence and Independence Vector Basis: Basis and its Dimension & Rank, Change of Basis, Null space and finding a basis for the null space, The Rank of a Matrix and Applications.	
3	**Eigenvalues, Eigenvectors and Diagonalization, Inner Product, Orthogonality**	**10**
	Eigenvalues, Eigenvectors and diagonalization: Eigenvalues and Eigenvectors, Existence of eigenvalues, Coordinate representation in terms of eigenvectors, Diagonalization Inner Product & Outer product: The inner product for vectors, Outer product for vectors Orthogonalization: Orthogonality, Projection orthogonal to multiple vectors, Projecting orthogonal to mutually orthogonal vectors	

PRACTICALS

Sr. No.	Topic.
1	Write a program to perform basic operations of complex number

2	Write a program to perform basic operations of vectors
3	Write a program to perform basic matrix operations
4	Write a program to inverse a matrix
5	Write a program to perform Gaussian Elimination
6	Write a program to convert any matrix to its echelon form
7	Write a program to convert any matrix to its row-canonical form
8	Write a program to enter a given matrix and an eigen value of the same. Find its eigen vector
9	Write a program to perform diagonalization of a matrix given its eigen values and eigen vector.
10	Write a program to do the following: 1. Enter a vector b and find the projection of b orthogonal to a given vector u. 2. Find the projection of b orthogonal to a set of given vectors

RECOMMENDED READING:

Text Books:

1. B. Kolman , D. Hill, Introductory Linear Algebra, An Applied First Course, Pearson Edn; 8th Edn; (2008)
2. Schaum's outlines Linear Algebra, Seymour Lipschutz, Marc Lars Lipson, 4th Edition, McGraw Hill

Reference Books

1. Linear Algebra and Probability for Computer Science Applications, Ernest Davis, A K Peters/CRC Press (2012).
2. Coding the Matrix Linear Algebra through Applications to Computer Science Edition 1, PHILIP N. KLEIN, Newtonian Press (2013)
3. Linear Algebra and Its Applications, Gilbert Strang, Cengage Learning, 4th Edition (2007).
4. Linear Algebra and Its Applications, David C Lay, Pearson Education India; 3rd Edition
5. H.Anton, Chris Rorres, Linear Algebra with Applns., Wiley, 7th Edn; (1994)

Program: B.Sc.- Computer Science					**Semester : IV**
Course: .NET Technologies					**Course Code: USMACS406**
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	02	-	2+1 = 3	25	75

Learning Objectives:

- To learn to create console applications using C#
- To explore .NET technologies for designing and developing dynamic websites
- To apply database connectivity in .Net Applications
- To create interactive and responsive web applications.

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Understand the .NET framework
CO2: Develop a proficiency in the C# programming language
CO3: Proficiently develop ASP.NET web applications using C#
CO4: Incorporate ADO.NET for data persistence in a web application

Outline of Syllabus: (per session plan)

Module	Description	No of hours
1	**The .NET Framework, C# Language Basics, ASP.NET**	10
2	**Web Controls, Validation, Master Pages**	10
3	**ADO.NET, Working with XML**	10
	Total	**30**
PRACTICALS		**30**

Module	.NET Technologies	No. of Hours/Credits 30/2
1	**The .NET Framework, C# Language Basics, ASP.NET**	10
	The .NET Framework: .NET Languages, Common Language Runtime, .NET Class Library **C# Language Basics**: Comments, Variables and Data Types, Variable Operations, Object-Based Manipulation, Conditional Logic, Loops, Methods, Classes, Value Types and Reference Types, Namespaces and Assemblies, Inheritance, Static Members, Casting Objects, Partial Classes **ASP.NET -** Writing Code - Code-Behind Class, Adding Event Handlers Anatomy of an ASP.NET Application - ASP.NET File Types, ASP.NET Web Folders **HTML Server Controls -** View State, HTML Control Classes, HTML Control Events, HtmlControl Base Class, HtmlContainerControl Class, HtmlInputControl Class, Page Class, global.asax File	
2	**Web Controls, Validation, Master Pages**	10
	Web Controls: Web Control Classes, WebControl Base Class, List Controls,Table Controls, Web Control Events and AutoPostBack, Page Life Cycle **State Management:** ViewState, Cross-Page Posting, Query String, Cookies,Session State, Configuring Session State, Application State **Validation:** Validation Controls, Server-Side Validation, ClientSide Validation, HTML5 Validation, Manual Validation, Validation with Regular Expressions **Rich Controls:** Calendar Control, AdRotator Control, MultiView Control ASP.NET Master Pages: Simple Master Page and Content Page, Connecting Master pages and Content Pages, Master Page with Multiple Content Regions, Master Pages and Relative Paths. Website Navigation: Site Maps, URL Mapping and Routing, SiteMapPath Control, TreeView Control, Menu Control	
3	**ADO.NET, Working with XML**	10
	ADO.NET: Data Provider Model, Direct Data Access - Creating a Connection, Select Command, DataReader, Disconnected Data Access **Data Binding :** Introduction, Single-Value Data Binding, Repeated-Value Data Binding, **Data Source Controls** – SqlDataSource Data Controls: GridView, DetailsView, FormView **Working with XML:** XML Classes – XMLTextWriter, XMLTextReader Caching: When to Use Caching, Output Caching, Data Caching	

PRACTICALS	
Sr. No.	**Topic.**
1	Write C# programs for understanding C# basics involving a. Variables and Data Types b. Object-Based Manipulation c. Conditional Logic d. Loops e. Methods
2	Write C# programs for Object oriented concepts of C# such as: a. Program using classes b. Constructor and Function Overloading c. Inheritance d. Namespaces
3	Design ASP.NET Pages with a. Server controls. b. Web controls and demonstrate the use of AutoPostBack c. Rich Controls (Calendar / Ad Rotator)
4	Design ASP.NET Pages for State Management using a. Cookies b. Session State c. Application State
5	Design ASP.NET page and perform validation using various Validation Controls
6	Design ASP.NET Pages with various Navigation Controls. Design an ASP.NET master web page and use it other (at least 2-3) content pages.
7	Perform ADO.NET data access in ASP.NET for Simple Data Binding
8	Perform ADO.NET data access in ASP.NET for Repeated Value Data Binding
9	Design ASP.NET application for Interacting (Reading / Writing) with XML documents
10	Design ASP.NET Pages for Performance improvement using Caching

RECOMMENDED READING:

Text Books:

1. Beginning ASP.NET 4.5 in C#, Matthew MacDonald, Apress(2012)

Reference Books

1. The Complete Reference ASP .NET, MacDonald, Tata McGraw Hill
2. Beginning ASP.NET 4 in C# and VB Imar Spanajaars, WROX

Program: B.Sc.- Computer Science					**Semester : IV**
Course: Software Engineering					**Course Code: USMACS407**
Teaching Scheme				**Evaluation Scheme**	
Lecture (Hours per week)	**Practical (Hours per week)**	**Tutorial (Hours per week)**	**Credit**	**Continuous Assessment and Evaluation (CAE) (Marks - 25)**	**Term End Examinations (TEE) (Marks-75 in Question Paper)**
02	--	-	2	25	75

Learning Objectives:

- To help students to develop skills that will enable them to construct software of high quality
- To make students understand how to develop software that is reliable, and that is reasonably easy to understand, modify and maintain.

Course Outcomes:

After completion of the course, learners would be able to:

CO1: Apply the software engineering lifecycle by demonstrating competence in various phases
CO2: Identify the defects and bugs that are present in the software and learn the ways to remove them
CO3: Understand different types of Software Testing techniques
CO4: Work in one or more significant application domains

Outline of Syllabus: (per session plan)

Module	**Description**	**No of hours**
1	Introduction, Project Feasibility Study	10
2	System Analysis, UML	10
3	Software Project Management, Software Measurement and Metrics	10
	Total	**30**
PRACTICALS		--

Module	Software Engineering	No. of Hours/Credits 30/2
1	**Introduction, Project Feasibility Study**	**10**
	Introduction: The Nature of Software, Software Engineering, The Software Process, Generic Process Model, The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Component-Based Development, The Unified Process Phases **Project Feasibility Study** - Operational, Technical, Economic, Organizational and Cultural feasibility. Defining project costs and project benefits. Cost/Benefit Analysis for a project	
2	**System Analysis, UML**	**10**
	Investigating System Requirements – Software Requirement Specification Document, Need of SRS, Characteristics & Components of SRS, Stakeholders, Identifying requirements using various techniques (such as Questionnaires, reviewing reports/forms, interviews, workflows etc) **UML:** Basics of UML, Types of UML Diagrams, Use Case Diagram, Class Diagram, Object Diagram, Sequence diagram & Collaboration diagram, State Transition & State chart diagrams UML Activity Diagram, Component Diagram, Package & Deployment Diagram System/Software Design, Architectural Design, Low-Level Design Coupling and Cohesion, Functional-Oriented Versus The Object-Oriented Approach, Design Specifications, Verification for Design, Monitoring and Control for Design	
3	**Software Project Management, Software Measurement and Metrics**	**10**
	Software Project Management: Estimation in Project Planning Process –Software Scope And Feasibility, Resource Estimation, Empirical Estimation Models – COCOMO II, Project Scheduling - Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-Line Charts **Software Measurement and Metrics:** Product Metrics – Measures, Metrics, and Indicators, Function-Based Metrics, Metrics for Object- Oriented Design, Metrics for Source Code, Halstead Metrics Applied to Testing, Metrics for Maintenance, Cyclomatic Complexity, Software Measurement - Size-Oriented,	

	Function-Oriented Metrics, Metrics for Software Quality **Software Testing :** Verification and Validation, Introduction to Testing, Testing Principles, Testing Objectives, Test Oracles, Levels of Testing, White-Box Testing/Structural Testing, Functional/Black-Box Testing, Test Plan, Test-Case Design	

RECOMMENDED READING:

Text Books:

1) Software Engineering, A Practitioner's Approach, Roger S, Pressman.(2014)

Reference Books

1) Software Engineering, Ian Sommerville, Pearson Education
2) Software Engineering: Principles and Practices, Deepak Jain,OXFORD University Press,
3) Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI
4) Software Engineering: Principles and Practices, Hans Van Vliet, John Wiley & Sons)
5) A Concise Introduction to Software Engineering, Pankaj Jalote, Springe