CRITERION I: CURRICULAR ASPECTS

1.3 – Curriculum Enrichment

1.3.2 - Number of courses that include experiential learning through project work/fieldwork/internship during the year

INDEX

B.E., CIVIL ENGINEERING

CE8501 DESIGN OF REINFORCED CEMENT CONCRETE ELEMENTS

OBJECTIVES:

To introduce the different types of philosophies related to design of basic structural elements such as slab, beam, column and footing which form part of any structural system with reference to Indian standard code of practice.

UNIT I INTRODUCTION

Objective of structural design-Steps in RCC Structural Design Process- Type of Loads on Structures and Load combinations- Code of practices and Specifications - Concept of Working Stress Method, Ultimate Load Design and Limit State Design Methods for RCC –Properties of Concrete and Reinforcing Steel - Analysis and Design of Singly reinforced Rectangular beams by working stress method - Limit State philosophy as detailed in IS code - Advantages of Limit State Method over other methods - Analysis and design of singly and doubly reinforced rectangular beams by Limit State Method.

UNIT II DESIGN OF BEAMS

Analysis and design of Flanged beams for – Use of design aids for Flexure - Behaviour of RC members in Shear, Bond and Anchorage - Design requirements as per current code - Behaviour of rectangular RC beams in shear and torsion - Design of RC members for combined Bending, Shear and Torsion.

UNIT III DESIGN OF SLABS AND STAIRCASE

Analysis and design of cantilever, one way simply supported and continuous slabs and supporting beams-Two way slab- Desingn of simply supported and continuous slabs using IS code coefficients- Types of Staircases – Design of dog-legged Staircase.

UNIT IV DESIGN OF COLUMNS

Types of columns –Axially Loaded columns – Design of short Rectangula Square and circular columns –Design of Slender columns- Design for Uniaxial and Biaxial bending using Column Curves

UNIT V DESIGN OF FOOTINGS

Concepts of Proportioning footings and foundations based on soil properties-Design of wall footing – Design of axially and eccentrically loaded Square, Rectangular pad and sloped footings – Design of Combined Rectangular footing for two columns only.

TOTAL: 75 PERIODS

OUTCOMES:

Students will be able to

- Understand the various design methodologies for the design of RC elements.
- Know the analysis and design of flanged beams by limit state method and sign of beams for shear, bond and torsion.
- design the various types of slabs and staircase by limit state method.
- Design columns for axial, uniaxial and biaxial eccentric loadings.
- Design of footing by limit state method.

CE8502 STRUCTURAL ANALYSIS I

OBJECTIVE:

To introduce the students to basic theory and concepts of classical methods of structural analysis

UNITI STRAIN ENERGY METHOD

Determination of Static and Kinematic Indeterminacies – Analysis of continuous beams, plane frames and indeterminate plane trusses by strain energy method (up to two degree of redundancy).

UNITII SLOPE DEFLECTION METHOD

Slope deflection equations – Equilibrium conditions - Analysis of continuous beams and rigid frames – Rigid frames with inclined members - Support settlements- symmetric frames with symmetric and skew-symmetric loadings.

UNITIII MOMENT DISTRIBUTION METHOD

Stiffness and carry over factors – Distribution and carryover of moments - Analysis of continuous Beams- Plane rigid frames with and without sway – Support settlement - symmetric frames with symmetric and skew-symmetric loadings.

UNITIV FLEXIBLITY METHOD

Primary structures - Compatibility conditions – Formation flexibility matrices - Analysis of indeterminate pin- jointed plane frames, continuous beams and rigid jointed plane frames by direct flexibility approach.

UNITV STIFFNESS METHOD

Restrained structure –Formation of stiffness matrices - equilibrium condition - Analysis of Continuous Beams, Pin-jointed plane frames and rigid frames by direct stiffness method.

TOTAL: 45 PERIODS

OUTCOMES:

Students will be able to

- Analyze continuous beams, pin-jointed indeterminate plane frames and rigid plane frames by strain energy method
- Analyse the continuous beams and rigid frames by slope defection method.
- Understand the concept of moment distribution and analysis of continuous beams and rigid frames with and without sway.
- Analyse the indeterminate pin jointed plane frames continuous beams and rigid frames using matrix flexibility method.
- Understand the concept of matrix stiffness method and analysis of continuous beams, pin jointed trusses and rigid plane frames.
- 4. Pandit G.S.andGupta S.P.,Structural Analysis–AMatrix Approach, Tata McGraw Hill Publishing Company Ltd.,2006

CE8703 STRUCTURAL DESIGN AND DRAWING

OBJECTIVE:

This course aims at providing students with a solid background on the principles of structural engineering design. Students will be acquire the knowledge of liquid retaining structures, bridges components, retaining wall and industrial structures.

UNIT I RETAINING WALLS

Reinforced concrete Cantilever and Counter fort Retaining Walls–Horizontal Backfill with Surcharge–Design of Shear Key-Design and Drawing.

UNIT II FLAT SLAB and BRIDGES

Design of Flat Slabs with and without drops by Direct Design Method of IS code- Design and Drawing - IRC Specifications and Loading – RC Solid Slab Bridge – Steel Foot-over Bridge-Design and Drawing.

UNIT III LIQUID STORAGE STRUCTURES

RCC Water Tanks - On ground, Elevated Circular, underground Rectangular Tanks–Hemispherical Bottomed Steel Water Tank –-Design and Drawing

UNIT IV INDUSTRIAL STRUCTURES

Structural steel Framing - Steel Roof Trusses – Roofing Elements – Beam columns – Codal provisions - Design and Drawing.

UNIT V GIRDERS AND CONNECTIONS

Plate Girders – Behaviour of Components-Deign of Welded Plate Girder-Design of Industrial Gantry Girders – Design of Eccentric Shear and Moment Resisting connections.

TOTAL: 75 PERIODS

Design and Drawing Exercises for practical component

Part A - RCC Structures

1. **Rectangular Column and Footing**
2. **Combined footing with Two columns**
3. **RCC one way &Two way Slab and beam system**
4. **Cantilever Retaining wall**
5. **RCC T beam bridge deck**
6. **Underground Rectangular Water Tank**
7. **Elevated circular water Tank**

Part B- Steel Structures

1. **Built up column, column base and Foundation**
2. **Simple Steel Roof Trusses**
3. **Industrial building Elements**
4. **Plate Girder (welded)**
5. **Framed Connections and Detailing**
6. **Gantry girder**
7. **Steel water Tank**

OUTCOMES:

At the end of the course the student will be able to

- Design and draw reinforced concrete Cantilever and Counterfort Retaining Walls
- Design and draw flat slab as per code provisions
- Design and draw reinforced concrete and steel bridges
- Design and draw reinforced concrete and steel water tanks
- Design and detail the various steel trusses and cantry girders

CE3403 **CONCRETE TECHNOLOGY**

COURSE OBJECTIVES:

- To study the properties of concrete making materials.
- To have better knowledge about the chemical and mineral admixtures in concrete.
- To familiarize with the IS method of mix design as per the latest code .
- To understand the fresh and hardened properties of concrete.
- To know the importance and applications of special concretes

UNIT I CONSTITUENT ATERIALS

Cement-Different types-Chemical composition and Properties -Tests on cement-IS Specifications- Aggregates-Classification-Mechanical properties and tests as per BIS Grading requirements-Water- Quality of water for use in concrete.

UNIT II CHEMICAL AND MINERAL ADMIXTURES

Accelerators-Retarders- Plasticisers- Super plasticizers- Water proofers - Mineral Admixtures like Fly Ash, Silica Fume, Ground Granulated Blast Furnace Slag and Metakaoline -Their effects on concrete properties

UNIT III PROPORTIONING OF CONCRETE MIX

Principles of Mix Proportioning-Properties of concrete related to Mix Design-Physical properties of materials required for Mix Design - Design Mix and Nominal Mix-BIS Method of Mix Design - Mix Design Examples

UNIT IV FRESH AND HARDENED PROPERTIES OF CONCRETE

Workability-Tests for workability of concrete-Slump Test and Compacting factor Test-Segregation and Bleeding-Determination of Compressive and Flexural strength as per BIS - Properties of Hardened concrete- Stress-strain curve for concrete-Determination of Modulus of elasticity.

UNIT V SPECIAL CONCRETES

Light weight concretes - High strength concrete - Fibre reinforced concrete – Ferrocement - Ready mix concrete - SIFCON - Shotcrete – Polymer concrete - High performance concrete- self compacting concrete - Geopolymer Concrete.

TOTAL : 45 PERIODS

COURSE OUTCOMES:

At the end of the course the student will be able to

1. **CO1** Understand the requirements of cement, aggregates and water for concrete
2. **CO2** Select suitable admixtures for enhancing the properties of concrete
3. **CO3** Design concrete mixes as per IS method of mix design
4. **CO4** Determine the properties of concrete at fresh and hardened state.
5. **CO5** Know the importance of special concretes for specific requirements.

CE3501 DESIGN OF REINFORCED CONCRETE STRUCTURAL ELEMENTS

COURSE OBJECTIVE:

To introduce the different design philosophy for reinforced concrete and discuss the limit state method of design of RC rectangular beams and to learn the concept in the design of RC flanged beams and design for shear and torsion and design of RC slabs and staircase, short RC columns, RC footing for walls, pad, sloped and combined rectangular footings.

UNIT I METHODS OF DESIGN OF CONCRETE STRUCTURES

Concept of Elastic method, ultimate load method and limit state method – Working stress method as detailed in IS code - Design of Singly Reinforced beam by working stress method - Limit State philosophy as detailed in IS code - Advantages of Limit State Method over other methods - Analysis and design of singly and doubly reinforced rectangular beams by limit State Method.

UNIT II LIMIT STATE METHOD - FLANGED BEAM, SHEAR & TORSION

Analysis and design of flanged beams – Use of design aids for Flexure - Behaviour of RC members in bond and Anchorage - Design requirements as per current code - Behaviour of RC beams in shear and torsion - Design of RC members for combined bending, shear and torsion - serviceability.

UNIT III LIMIT STATE DESIGN OF SLABS AND STAIRCASE

Analysis and design of cantilever, one way, two way and continuous slabs subjected to uniformly distributed load for various boundary conditions- Types of Staircases – Design of dog-legged Staircase –Introduction to Flat Slab.

UNIT IV LIMIT STATE DESIGN OF COLUMNS

Types of columns – Design of short Rectangular and circular columns for axial, uniaxial and biaxial bending.

UNIT V LIMIT STATE DESIGN OF FOOTING

Design of wall footing – Design of axially and eccentrically loaded rectangular pad and sloped footings – Design of combined rectangular footing for two columns only.

TOTAL: 45 PERIODS

COURSE OUTCOMES:

At the end of the course the student will be able to

1. **CO1** Know the various design concepts and design RC rectangular beams by working stress and limit state methods
2. **CO2** Understand the design of flanged beams, design for shear and torsion, and anchorage and development length.
3. **CO3** Design a RC slabs and staircase and draw the reinforcement detailing.
4. **CO4** Design short columns for axial, uni-axial and bi-axial eccentric loadings
5. **CO5** Design wall footings, isolated footings and combined rectangular footing.

B.E., COMPUTER SCIENCE AND ENGINEERING

CS3391 OBJECT ORIENTED PROGRAMMING

UNIT I INTRODUCTION TO OOP AND JAVA

Overview of OOP – Object oriented programming paradigms – Features of Object Oriented Programming – Java Buzzwords – Overview of Java – Data Types, Variables and Arrays – Operators – Control Statements – Programming Structures in Java – Defining classes in Java – Constructors-Methods -Access specifiers - Static members- Java Doc comments

UNIT II INHERITANCE, PACKAGES AND INTERFACES

Overloading Methods – Objects as Parameters – Returning Objects –Static, Nested and Inner Classes. Inheritance: Basics– Types of Inheritance -Super keyword -Method Overriding – Dynamic Method Dispatch –Abstract Classes – final with Inheritance. Packages and Interfaces: Packages –Packages and Member Access –Importing Packages – Interfaces.

UNIT III EXCEPTION HANDLING AND MULTITHREADING

Exception Handling basics – Multiple catch Clauses – Nested try Statements – Java's Built-in Exceptions – User defined Exception. Multithreaded Programming: Java Thread Model–Creating a Thread and Multiple Threads – Priorities – Synchronization – Inter Thread Communication-Suspending –Resuming, and Stopping Threads –Multithreading. Wrappers – Auto boxing.

UNIT IV I/O, GENERICS, STRING HANDLING

I/O Basics – Reading and Writing Console I/O – Reading and Writing Files. Generics: Generic Programming – Generic classes – Generic Methods – Bounded Types – Restrictions and Limitations. Strings: Basic String class, methods and String Buffer Class.

UNIT V JAVAFX EVENT HANDLING, CONTROLS AND COMPONENTS

JAVAFX Events and Controls: Event Basics – Handling Key and Mouse Events. Controls: Checkbox, ToggleButton – RadioButtons – ListView – ComboBox – ChoiceBox – Text Controls –ScrollPane. Layouts – FlowPane – HBox and VBox – BorderPane – StackPane – GridPane. Menus – Basics – Menu – Menu bars – MenuItem.

COURSE OUTCOMES:

On completion of this course, the students will be able to

CO1:Apply the concepts of classes and objects to solve simple problems

CO2:Develop programs using inheritance, packages and interfaces

CO3:Make use of exception handling mechanisms and multithreaded model to solve real world problems

CO4:Build Java applications with I/O packages, string classes, Collections and generics concepts

CO5:Integrate the concepts of event handling and JavaFX components and controls for developing GUI based applications.

CCS356 OBJECT ORIENTED SOFTWARE ENGINEERING

UNIT I SOFTWARE PROCESS AND AGILE DEVELOPMENT

Introduction to Software Engineering, Software Process, Perspective and Specialized Process Models –Introduction to Agility-Agile process-Extreme programming-XP Process-Case Study.

UNIT II REQUIREMENTS ANALYSIS AND SPECIFICATION

Requirement analysis and specification – Requirements gathering and analysis – Software Requirement Specification – Formal system specification – Finite State Machines – Petrinets – Object modelling using UML – Use case Model – Class diagrams – Interaction diagrams – Activity diagrams – State chart diagrams – Functional modelling – Data Flow Diagram- CASE TOOLS.

UNIT III SOFTWARE DESIGN

Software design – Design process – Design concepts – Coupling – Cohesion – Functional independence – Design patterns – Model-view-controller – Publish-subscribe – Adapter – Command – Strategy – Observer – Proxy – Facade – Architectural styles – Layered - Client Server - Tiered -Pipe and filter- User interface design-Case Study.

UNIT IV SOFTWARE TESTING AND MAINTENANCE 9

Testing – Unit testing – Black box testing– White box testing – Integration and System testing– Regression testing – Debugging - Program analysis – Symbolic execution – Model Checking-Case Study

UNIT V PROJECT MANAGEMENT

Software Project Management- Software Configuration Management - Project Scheduling-DevOps: Motivation-Cloud as a platform-Operations- Deployment Pipeline:Overall Architecture Building and Testing-Deployment- Tools- Case Study

COURSE OUTCOMES:

CO1: Compare various Software Development Lifecycle Models
CO2: Evaluate project management approaches as well as cost and schedule estimation strategies.
CO3: Perform formal analysis on specifications.
CO4: Use UML diagrams for analysis and design.
CO5: Architect and design using architectural styles and design patterns, and test the system

CCS375 WEB TECHNOLOGIES

COURSE OBJECTIVES:

- To understand different Internet Technologies
- To learn java-specific web services architecture
- To Develop web applications using frameworks

UNIT I WEBSITE BASICS, HTML 5, CSS 3, WEB 2.0 7

Web Essentials: Clients, Servers and Communication – The Internet – World wide web – HTTP Request Message – HTTP Response Message – Web Clients – Web Servers – HTML5 – Tables –Lists – Image – HTML5 control elements – Drag and Drop – Audio – Video controls - CSS3 – Inline, embedded and external style sheets – Rule cascading – Inheritance – Backgrounds – Border Images – Colors – Shadows – Text – Transformations – Transitions – Animations. Bootstrap Framework

UNIT II CLIENT SIDE PROGRAMMING

Java Script: An introduction to JavaScript–JavaScript DOM Model-Exception Handling-Validation-Built-in objects-Event Handling- DHTML with JavaScript- JSON introduction – Syntax – Function Files.

UNIT III SERVER SIDE PROGRAMMING

Servlets: Java Servlet Architecture- Servlet Life Cycle- Form GET and POST actions- Session Handling- Understanding Cookies- DATABASE CONNECTIVITY: JDBC.

UNIT IV PHP and XML

An introduction to PHP: PHP- Using PHP- Variables- Program control- Built-in functions- Form Validation. XML: Basic XML- Document Type Definition- XML Schema, XML Parsers and Validation,
XSL

UNIT V INTRODUCTION TO ANGULAR and WEB APPLICATIONS FRAMEWORKS

Introduction to AngularJS, MVC Architecture, Understanding ng attributes, Expressions and data binding, Conditional Directives, Style Directives, Controllers, Filters, Forms, Routers, Modules, Services; Web Applications Frameworks and Tools – Firebase- Docker- Node JS-React- Django-UI & UX.

CCS332 APP DEVELOPMENT

COURSE OBJECTIVES:

- To learn development of native applications with basic GUI Components
- To develop cross-platform applications with event handling
- To develop applications with location and data storage capabilities
- To develop web applications with database access

UNIT I FUNDAMENTALS OF MOBILE & WEB APPLICATION DEVELOPMENT

Basics of Web and Mobile application development, Native App, Hybrid App, Cross-platform App, What is Progressive Web App, Responsive Web design.

UNIT II NATIVE APP DEVELOPMENT USING JAVA

Native Web App, Benefits of Native App, Scenarios to create Native App, Tools for creating Native App, Cons of Native App, Popular Native App Dev elopment Frameworks, Java & Kotlin for Android, Swift & Objective-C for iOS, Basics of React Native, Native Components, JSX, State, Props

UNIT III HYBRID APP DEVELOPMENT

Hybrid Web App, Benefits of Hybrid App, Criteria for creating Native App, Tools for creating Hybrid App, Cons of Hybrid App, Popular Hybrid App Development Frameworks, Ionic, Apache
Cordova

UNIT IV CROSS-PLATFORM APP DEVELOPMENT USING REACT-NATIVE

What is Cross-platform App, Benefits of Cross-platform App, Criteria for creating Cross-platform App, Tools for creating Cross-platform App, Cons of Cross-platform App, Popular Cross-platform App Development Frameworks, Flutter, Xamarin, React-Native, Basics of React Native, Native Components, JSX, State, Props

UNIT V NON-FUNCTIONAL CHARACTERISTICS OF APP FRAMEWORKS

Comparison of different App frameworks, Build Performance, App Performance, Debugging capabilities, Time to Market, Maintainability, Ease of Development, UI/UX, Reusability

CCS370 UI AND UX DESIGN

COURSE OBJECTIVES:

- To provide a sound knowledge in UI & UX
- To understand the need for UI and UX
- To understand the various Research Methods used in Design
- To explore the various Tools used in UI & UX
- Creating a wireframe and prototype

UNIT I FOUNDATIONS OF DESIGN

UI vs. UX Design - Core Stages of Design Thinking - Divergent and Convergent Thinking - Brainstorming and Game storming - Observational Empathy

UNIT II FOUNDATIONS OF UI DESIGN

Visual and UI Principles - UI Elements and Patterns - Interaction Behaviors and Principles – Branding - Style Guides

UNIT III FOUNDATIONS OF UX DESIGN

Introduction to User Experience - Why You Should Care about User Experience - Understanding User Experience - Defining the UX Design Process and its Methodology - Research in User Experience Design - Tools and Method used for Research - User Needs and its Goals - Know about Business Goals

UNIT IV WIREFRAMING, PROTOTYPING AND TESTING

Sketching Principles - Sketching Red Routes - Responsive Design – Wireframing - Creating Wireflows - Building a Prototype - Building High-Fidelity Mockups - Designing Efficiently with Tools - Interaction Patterns - Conducting Usability Tests - Other Evaluative User Research Methods -Synthesizing Test Findings - Prototype Iteration

UNIT V RESEARCH, DESIGNING, IDEATING, & INFORMATION ARCHITECTURE

Identifying and Writing Problem Statements - Identifying Appropriate Research Methods - Creating Personas - Solution Ideation - Creating User Stories - Creating Scenarios - Flow Diagrams - Flow Mapping - Information Architecture

CCS342 DEVOPS

UNIT I INTRODUCTION TO DEVOPS

Devops Essentials - Introduction To AWS, GCP, Azure - Version control systems: Git and Github.

UNIT II COMPILE AND BUILD USING MAVEN & GRADLE

Introduction, Installation of Maven, POM files, Maven Build lifecycle, Build phases(compile build, test, package) Maven Profiles, Maven repositories(local, central, global),Maven plugins, Maven create and build Artificats, Dependency management, Installation of Gradle, Understand build using Gradle

UNIT III CONTINUOUS INTEGRATION USING JENKINS

Install & Configure Jenkins, Jenkins Architecture Overview, Creating a Jenkins Job, Configuring a Jenkins job, Introduction to Plugins, Adding Plugins to Jenkins, Commonly used plugins (Git Plugin, Parameter Plugin, HTML Publisher, Copy Artifact and Extended choice parameters). Configuring Jenkins to work with java, Git and Maven, Creating a Jenkins Build and Jenkins workspace.

UNIT IV CONFIGURATION MANAGEMENT USING ANSIBLE

Ansible Introduction, Installation, Ansible master/slave configuration, YAML basics, Ansible modules, Ansible Inventory files, Ansible playbooks, Ansible Roles, adhoc commands in ansible

UNIT V BUILDING DEVOPS PIPELINES USING AZURE

Create Github Account, Create Repository, Create Azure Organization, Create a new pipeline, Build a sample code, Modify azure-pipelines.yaml file

CCS332 APP DEVELOPMENT

COURSE OBJECTIVES:

- To learn development of native applications with basic GUI Components
- To develop cross-platform applications with event handling
- To develop applications with location and data storage capabilities
- To develop web applications with database access

UNIT I FUNDAMENTALS OF MOBILE & WEB APPLICATION DEVELOPMENT

Basics of Web and Mobile application development, Native App, Hybrid App, Cross-platform App, What is Progressive Web App, Responsive Web design

UNIT II NATIVE APP DEVELOPMENT USING JAVA

Native Web App, Benefits of Native App, Scenarios to create Native App, Tools for creating Native App, Cons of Native App, Popular Native App Dev elopment Frameworks, Java & Kotlin for Android, Swift & Objective-C for iOS, Basics of React Native, Native Components, JSX, State, Props

UNIT III HYBRID APP DEVELOPMENT

Hybrid Web App, Benefits of Hybrid App, Criteria for creating Native App, Tools for creating Hybrid App, Cons of Hybrid App, Popular Hybrid App Development Frameworks, Ionic, Apache
Cordova

UNIT IV CROSS-PLATFORM APP DEVELOPMENT USING REACT-NATIVE

What is Cross-platform App, Benefits of Cross-platform App, Criteria for creating Cross-platform App, Tools for creating Cross-platform App, Cons of Cross-platform App, Popular Cross-platform App Development Frameworks, Flutter, Xamarin, React-Native, Basics of React Native, Native Components, JSX, State, Props

UNIT V NON-FUNCTIONAL CHARACTERISTICS OF APP FRAMEWORKS

Comparison of different App frameworks, Build Performance, App Performance, Debugging capabilities, Time to Market, Maintainability, Ease of Development, UI/UX, Reusability

COURSE OUTCOMES:

CO1:Develop Native applications with GUI Components.
CO2:Develop hybrid applications with basic event handling.
CO3: Implement cross-platform applications with location and data storage capabilities.
CO4: Implement cross platform applications with basic GUI and event handling.
CO5:Develop web applications with cloud database access.

CS3491 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

COURSE OBJECTIVES:

- The main objectives of this course are to:
- Study about uninformed and Heuristic search techniques.
- Learn techniques for reasoning under uncertainty
- Introduce Machine Learning and supervised learning algorithms
- Study about ensembling and unsupervised learning algorithms
- Learn the basics of deep learning using neural networks

UNIT I PROBLEM SOLVING

Introduction to AI - AI Applications - Problem solving agents – search algorithms – uninformed search strategies – Heuristic search strategies – Local search and optimization problems – adversarial search – constraint satisfaction problems (CSP)

UNIT II PROBABILISTIC REASONING

Acting under uncertainty – Bayesian inference – naïve bayes models. Probabilistic reasoning – Bayesian networks – exact inference in BN – approximate inference in BN – causal networks.

UNIT III SUPERVISED LEARNING

Introduction to machine learning – Linear Regression Models: Least squares, single & multiple variables, Bayesian linear regression, gradient descent, Linear Classification Models: Discriminant function – Probabilistic discriminative model - Logistic regression, Probabilistic generative model –Naive Bayes, Maximum margin classifier – Support vector machine, Decision Tree, Random forests

UNIT IV ENSEMBLE TECHNIQUES AND UNSUPERVISED LEARNING

Combining multiple learners: Model combination schemes, Voting, Ensemble Learning - bagging, boosting, stacking, Unsupervised learning: K-means, Instance Based Learning: KNN, Gaussian mixture models and Expectation maximization

UNIT V NEURAL NETWORKS

Perceptron - Multilayer perceptron, activation functions, network training – gradient descent optimization – stochastic gradient descent, error backpropagation, from shallow networks to deep networks –Unit saturation (aka the vanishing gradient problem) – ReLU, hyperparameter tuning, batch normalization, regularization, dropout.

CS3352 FOUNDATIONS OF DATA SCIENCE

COURSE OBJECTIVES:

- To understand the data science fundamentals and process.
- To learn to describe the data for the data science process.
- To learn to describe the relationship between data.
- To utilize the Python libraries for Data Wrangling.
- To present and interpret data using visualization libraries in Python

UNIT I INTRODUCTION

Data Science: Benefits and uses – facets of data - Data Science Process: Overview – Defining research goals – Retrieving data – Data preparation - Exploratory Data analysis – build the model–presenting findings and building applications - Data Mining - Data Warehousing – Basic Statistical descriptions of Data

UNIT II DESCRIBING DATA

Types of Data - Types of Variables -Describing Data with Tables and Graphs –Describing Data with Averages - Describing Variability - Normal Distributions and Standard (z) Scores

UNIT III DESCRIBING RELATIONSHIPS

Correlation –Scatter plots –correlation coefficient for quantitative data –computational formula for correlation coefficient – Regression –regression line –least squares regression line – Standard error of estimate – interpretation of r2 –multiple regression equations –regression towards the mean

UNIT IV PYTHON LIBRARIES FOR DATA WRANGLING

Basics of Numpy arrays –aggregations –computations on arrays –comparisons, masks, boolean logic – fancy indexing – structured arrays – Data manipulation with Pandas – data indexing and selection – operating on data – missing data – Hierarchical indexing – combining datasets – aggregation and grouping – pivot tables

UNIT V DATA VISUALIZATION

Importing Matplotlib – Line plots – Scatter plots – visualizing errors – density and contour plots –Histograms – legends – colors – subplots – text and annotation – customization – three dimensional plotting - Geographic Data with Basemap - Visualization with Seaborn.

GE3151 PROBLEM SOLVING AND PYTHON PROGRAMMING

COURSE OBJECTIVES:

- To understand the basics of algorithmic problem solving.
- To learn to solve problems using Python conditionals and loops.
- To define Python functions and use function calls to solve problems.
- To use Python data structures - lists, tuples, dictionaries to represent complex data.
- To do input/output with files in Python.

UNIT I COMPUTATIONAL THINKING AND PROBLEM SOLVING

Fundamentals of Computing – Identification of Computational Problems -Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi.

UNIT II DATA TYPES, EXPRESSIONS, STATEMENTS

Python interpreter and interactive mode,debugging; values and types: int, float, boolean, string , and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.

UNIT III CONTROL FLOW, FUNCTIONS, STRINGS

Conditionals:Boolean values and operators, conditional (if), alternative (if-else),chained conditional (if-elif-else);Iteration: state, while, for, break, continue, pass; Fruitful functions: return values,parameters, local and global scope, function composition, recursion; Strings: string slices,immutability, string functions and methods, string module; Lists as arrays. Illustrative programs: square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.

UNIT IV LISTS, TUPLES, DICTIONARIES

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension; Illustrative programs: simple sorting, histogram, Students marks statement, Retail bill preparation.

UNIT V FILES, MODULES, PACKAGES

Files and exceptions: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file, Voter's age validation, Marks range validation (0-100).

CCS340 CYBER SECURITY

COURSE OBJECTIVES:

- To learn cybercrime and cyberlaw.
- To understand the cyber attacks and tools for mitigating them.
- To understand information gathering.
- To learn how to detect a cyber attack.
- To learn how to prevent a cyber attack.

UNIT I INTRODUCTION

Cyber Security – History of Internet – Impact of Internet – CIA Triad; Reason for Cyber Crime – Need for Cyber Security – History of Cyber Crime; Cybercriminals – Classification of Cybercrimes –A Global Perspective on Cyber Crimes; Cyber Laws – The Indian IT Act – Cybercrime and Punishment.

UNIT II ATTACKS AND COUNTERMEASURES

OSWAP; Malicious Attack Threats and Vulnerabilities: Scope of Cyber-Attacks – Security Breach –Types of Malicious Attacks – Malicious Software – Common Attack Vectors – Social engineering Attack – Wireless Network Attack – Web Application Attack – Attack Tools – Countermeasures.

UNIT III RECONNAISSANCE

Harvester – Whois – Netcraft – Host – Extracting Information from DNS – Extracting Information from E-mail Servers – Social Engineering Reconnaissance; Scanning – Port Scanning – Network Scanning and Vulnerability Scanning – Scanning Methodology – Ping Sweer Techniques – Nmap Command Switches – SYN – Stealth – XMAS – NULL – IDLE – FIN Scans – Banner Grabbing and OS Finger printing Techniques.

UNIT IV INTRUSION DETECTION

Host -Based Intrusion Detection – Network -Based Intrusion Detection – Distributed or Hybrid Intrusion Detection – Intrusion Detection Exchange Format – Honeypots – Example System Snort.

UNIT V INTRUSION PREVENTION

Firewalls and Intrusion Prevention Systems: Need for Firewalls – Firewall Characteristics and Access Policy – Types of Firewalls – Firewall Basing – Firewall Location and Configurations – Intrusion Prevention Systems – Example Unified Threat Management Products.

GE3754 HUMAN RESOURCE MANAGEMENT

COURSE OBJECTIVE:

- To provide knowledge about management issues related to staffing,
- To provide knowledge about management issues related to training,
- To provide knowledge about management issues related to performance
- To provide knowledge about management issues related to compensation
- To provide knowledge about management issues related to human factors consideration
- and compliance with human resource requirements.

UNIT I INTRODUCTION TO HUMAN RESOURCE MANAGEMENT

The importance of human resources – Objective of Human Resource Management - Human resource policies - Role of human resource manager.

UNIT II HUMAN RESOURCE PLANNING

Importance of Human Resource Planning – Internal and External sources of Human Resources - Recruitment - Selection – Socialization.

UNIT III TRAINING AND EXECUTIVE DEVELOPMENT

Types of training and Executive development methods – purpose – benefits.

UNIT IV EMPLOYEE COMPENSATION

Compensation plan – Reward – Motivation – Career Development - Mentor – Protege relationships.

UNIT V PERFORMANCE EVALUATION AND CONTROL

Performance evaluation – Feedback - The control process – Importance – Methods – grievances –Causes – Redressal methods.

COURSE OUTCOMES:

CO1: Students would have gained knowledge on the various aspects of HRM
CO2: Students will gain knowledge needed for success as a human resources professional.
CO3: Students will develop the skills needed for a successful HR manager.
CO4: Students would be prepared to implement the concepts learned in the workplace.
CO5: Students would be aware of the emerging concepts in the field of HRM

OMG355 MULTIVARIATE DATA ANALYSIS

COURSE OBJECTIVE:

- To know various multivariate data analysis techniques for business research.

UNIT I INTRODUCTION

Uni-variate, Bi-variate and Multi-variate techniques – Classification of multivariate techniques – Guidelines for multivariate analysis and interpretation.

UNIT II PREPARING FOR MULTIVARIATE ANALYSIS

Conceptualization of research model with variables, collection of data –-Approaches for dealing with missing data – Testing the assumptions of multivariate analysis.

UNIT III MULTIPLE LINEAR REGRESSION ANALYSIS, FACTOR ANALYSIS

Multiple Linear Regression Analysis – Inferences from the estimated regression function – Validation of the model. -Approaches to factor analysis – interpretation of results.

UNIT IV LATENT VARIABLE TECHNIQUES

Confirmatory Factor Analysis, Structural equation modelling, Mediation models, Moderation models, Longitudinal studies.

UNIT V ADVANCED MULTIVARIATE TECHNIQUES

Multiple Discriminant Analysis, Logistic Regression, Cluster Analysis, Conjoint Analysis, multidimensional scaling.

B.E., ELECTRICAL AND ELECTRONICS ENGINEERING

EE8691 EMBEDDED SYSTEMS

COURSE OBJECTIVES:

- To impart knowledge on the following Topics
- Building Blocks of Embedded System
- Various Embedded Development Strategies
- Bus Communication in processors, Input/output interfacing.
- Various processor scheduling algorithms.
- Basics of Real time operating system and example tutorials to discuss on one real timeoperating system tool.

UNIT I INTRODUCTION TO EMBEDDED SYSTEMS

Introduction to Embedded Systems –Structural units in Embedded processor , selection of processor & memory devices- DMA — Memory management methods- Timer and Counting devices, Watchdog Timer, Real Time Clock, In circuit emulator, Target Hardware Debugging.

UNIT II EMBEDDED NETWORKING

Embedded Networking: Introduction, I/O Device Ports & Buses– Serial Bus communication protocols RS232 standard – RS422 – RS 485 - CAN Bus -Serial Peripheral Interface (SPI) – InterIntegrated Circuits (I^2C) –need for device drivers.

UNIT III EMBEDDED FIRMWARE DEVELOPMENT ENVIRONMENT

Embedded Product Development Life Cycle- objectives, different phases of EDLC, Modelling of EDLC; issues in Hardware-software Co-design, Data Flow Graph, state machine model, Sequential Program Model, concurrent Model, object oriented Model.

UNIT IV RTOS BASED EMBEDDED SYSTEM DESIGN

Introduction to basic concepts of RTOS- Task, process & threads, interrupt routines in RTOS, Multiprocessing and Multitasking, Preemptive and non-preemptive scheduling, Task communication shared memory, message passing-, Inter process Communication — synchronization between processes-semaphores, Mailbox, pipes, priority inversion, priority inheritance.

UNIT V EMBEDDED SYSTEM APPLICATION AND DEVELOPMENT

Case Study of Washing Machine- Automotive Application- Smart card System Application-ATM machine –Digital camera

COURSE OUTCOMES:

- Ability to understand and analyze Embedded systems.
- Ability to suggest an embedded system for a given application.
- Ability to operate various Embedded Development Strategies
- Ability to study about the bus Communication in processors.
- Ability to acquire knowledge on various processor scheduling algorithms.
- Ability to understand basics of Real time operating system.

GE3151 PROBLEM SOLVING AND PYTHON PROGRAMMING

COURSE OBJECTIVES:

- To understand the basics of algorithmic problem solving.
- To learn to solve problems using Python conditionals and loops.
- To define Python functions and use function calls to solve problems.
- To use Python data structures - lists, tuples, dictionaries to represent complex data.
- To do input/output with files in Python.

UNIT I COMPUTATIONAL THINKING AND PROBLEM SOLVING

Fundamentals of Computing – Identification of Computational Problems -Algorithms, building blocks ofalgorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi.

UNIT II DATA TYPES, EXPRESSIONS, STATEMENTS

Python interpreter and interactive mode, debugging; values and types: int, float, boolean, string, and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.

UNIT III CONTROL FLOW, FUNCTIONS, STRINGS

Conditionals: Boolean values and operators, conditional (if), alternative (if-else), chained conditional(if-elif-else); Iteration: state, while, for, break, continue, pass; Fruitful functions: return values,parameters, local and global scope, function composition, recursion; Strings string slices, immutability, string functions and methods, string module; Lists as arrays. Illustrative programs:square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.

UNIT IV LISTS, TUPLES, DICTIONARIES

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension; Illustrative programs: simple sorting, histogram, Students marks statement, Retail bill preparation.

UNIT V FILES, MODULES, PACKAGES

Files and exception: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file, Voter's age validation, Marks range validation (0-100).

COURSE OUTCOMES:

Upon completion of the course, students will be able to

CO1: Develop algorithmic solutions to simple Computational problems.
CO2: Develop and execute simple Python programs.
CO3: Write simple Python programs using conditionals and looping for solving problems.
CO4: Decompose a Python program into functions.
CO5: Represent compound data using Python lists, tuples, dictionaries etc.
CO6: Read and write data from/to files in Python programs.

B.E., ELECTRONICS AND COMMUNICATION ENGINEERING

ET3491 EMBEDDED SYSTEMS AND IOT DESIGN

COURSE OBJECTIVES:

- Learn the architecture and features of 8051.
- Study the design process of an embedded system.
- Understand the real–time processing in an embedded system.
- Learn the architecture and design flow of IoT.
- Build an IoT based system.

UNIT I 8051 MICROCONTROLLER

Microcontrollers for an Embedded System – 8051 – Architecture – Addressing Modes – Instruction Set – Program and Data Memory – Stacks – Interrupts – Timers/Counters – Serial Ports – Programming.

UNIT II EMBEDDED SYSTEMS

Embedded System Design Process – Model Train Controller – ARM Processor – Instruction Set Preliminaries – CPU – Programming Input and Output – Supervisor Mode – Exceptions and Trap – Models for programs – Assembly, Linking and Loading – Compilation Techniques – Program Level Performance Analysis.

UNIT III PROCESSES AND OPERATING SYSTEMS

Structure of a real – time system – Task Assignment and Scheduling – Multiple Tasks and Multiple Processes – Multirate Systems – Pre emptive real – time Operating systems – Priority based scheduling – Interprocess Communication Mechanisms – Distributed Embedded Systems – MPSoCs and Shared Memory Multiprocessors – Design Example – Audio Player, Engine Control Unit and Video Accelerator.

UNIT IV IOT ARCHITECTURE AND PROTOCOLS

Internet – of – Things – Physical Design, Logical Design – IoT Enabling Technologies – Domain Specific IoTs – IoT and M2M – IoT System Management with NETCONF – YANG – IoT Platform Design – Methodology – IoT Reference Model – Domain Model – Communication Model – IoT Reference Architecture – IoT Protocols -MQTT, XMPP, Modbus, CANBUS and BACNet.

UNITV IOT SYSTEM DESIGN

Basicbuilding blocks of an IoT device –RaspberryPi–Board–Linuxon Raspberry Pi– Interfaces – Programming with Python – Case Studies: Home Automation, Smart Cities, Environment and Agriculture.

COURSEOUTCOMES:

CO1: Explain the architecture and features of 8051.
CO2: Develop a model of an embedded system.
CO3: List the concepts of real time operating systems.
CO4: Learn the architecture and protocols of IoT.
CO5: Design an IoT based system for any application.

CEC365 WIRELESS SENSOR NETWORK DESIGN

COURSE OBJECTIVES:

- To understand the fundamentals of wireless sensor network
- To gain knowledge on the MAC and Routing Protocols of WSN
- To get exposed to 6 LOWPAN technology
- To acquire knowledge on the protocols required for developing real time applications using WSN and 6LOWPAN.
- To gain knowledge about operating system related to WSN and 6 LOWPAN

UNIT I INTRODUCTION

Principle of Wireless Sensor Network -Introduction to wireless sensor networks-Challenges, Comparison with ad hoc network, Node architecture and Network architecture, design principles, Service interfaces,Gateway, Short range radio communication standards-IEEE802.15.4,Zigbee and Bluetooth. Physical layer and transceiver design considerations.

UNIT II MAC AND ROUTING PROTOCOLS

MAC protocols – fundamentals, low duty cycle protocols and wakeup concepts, contention and Schedule-based protocols -SMAC, BMAC,TRAMA, Routing protocols – Requirements,ClassificatioN-SPIN, Directed Diffusion, COUGAR, ACQUIRE, LEACH, PEGASIS.

UNIT III 6 LOWPAN

6LoWPAN Architecture - protocol stack, Adaptation Layer, Link layers – Addressing, Routing - Mesh- Under - Route-Over, Header Compression - Stateless header compression - Context- based header compression, Fragmentation and Reassembly , Mobility – types, Mobile IPv6, Proxy Home Agent, Proxy MIPv6, NEMO –Routing – MANET, ROLL, Border routing.

UNIT IV APPLICATION

Design issued, protocol paradigms- ene to end, real time streaming and sessions publish/subscribe, Web service paradigms, Common Protocols -Web service protocols, MQtelemetry transport for sensor networks (MQTT-S), ZigBee compact application protocol (CAP),Service discovery, Simple network management protocol (SNMP), Real-time transport and sessions, Industry- Specific protocols.

UNIT V TOOLS

TinyOS – Introduction, NesC, Interfaces, modules, configuration, Programming in TinyOS using NesC, TOSSIM, Contiki – Structure, Communication Stack, Simulation environment – Cooja simulator, Programming

COURSE OUTCOMES:

CO1: To be able to design solutions for WSNs applications

CO2: To be able to develop efficient MAC and Routing Protocols

CO3: To be able to design solutions for 6LOWPAN applications

CO4: To be able to develop efficient layered protocols in 6 LOWPAN

CO5:To be able to use Tiny OS and Contiki Osin WSNs and 6 LOWPAN applications

CEC367 INDUSTRIAL IOT AND INDUSTRY 4.0

COURSE OBJECTIVES:

- IoT Nodes & Sensors
- IoT Gateways
- IoT Cloud Systems
- IoT Cloud Dashboards
- Challenges in IoT system Design–Hardware & Software

UNIT I UNDERSTANDING IOT CONCEPT AND DEVELOPMENT PLATFORM

IOT Definition, Importance of IoT, Applications of IOT, IoT architecture, Understanding working ofSensors, Actuators, Sensor calibration, Study of Different sensors and their characteristics

UNIT II ANALYZING & DECODING OF COMMUNICATION PROTOCOL USED IN IOT DEVELOPMENT PLATFORM

UART Communication Protocol, I2C Protocol device interfacing and decoding of signal, SPIProtocol device interfacing and decoding of signal, WIFI and Router interfacing, Ethernet Configuration, Bluetooth study and analysis of data flow, Zigbee Interfacing and study of signalflow

UNIT III IOT PHYSICAL DEVICES AND END POINTS AND CONTROLLING HARDWARE AND SENSORS

IoT Physical Devices and Endpoints- Introduction to Arduino and Raspberry Pi- Installation, Interfaces (serial, SPI, I2C), Programming – Python program with Raspberry PI with focus on interfacing external gadgets, controlling output, reading input from pins.

Controlling Hardware- Connecting LED, Buzzer, Switching High Power devices with transistors, Controlling AC Power devices with Relays, Controlling servo motor, speed control of DC Motor, unipolar and bipolar Stepper motors;

Sensors- Light sensor, temperature sensor with thermistor, voltage sensor, ADC and DAC, Temperature and Humidity Sensor DHT11, Motion Detection Sensors, Wireless Bluetooth Sensors,Level Sensors, USB Sensors, Embedded Sensors, Distance Measurement with ultrasound sensor.

UNIT IV CLOUD SERVICES USED IN IOT DEVELOPMENT PLATFORM

Configuration of the cloud platform, Sending data from the IOT nodes to the gateways usingdifferent communication options; Transferring data from gateway to the cloud; Exploring the web services like mail, Messaging (SMS) and Twitter etc.;Tracking of cloud data as per the requirement; Google Cloud service architect; AWS clod Services architect; Microsoft Azure cloud services Architect; OEN source Cloud Services; Initial State Iot Dashboard & Cloud Services

UNIT V CHALLENGES IN IOT SYSTEM DESIGN–HARDWARE & SOFTWARE

Antenna design and placement, Chip-package system development, Power electronics, electromagnetic interference/compatibility (EMI/EMC), Electronics reliability; Battery simulation.

COURSE OUTCOMES:

Upon completion of this course, the students will be able to

CO1:Understand the building blocks of IoT technology and explore the vast spectrum of IoT applications

CO2: Use processors & peripherals to design & build IoT hardware

CO3: Assess, select and customize technologies for IoT applications

CO4: Connect numerous IOT applications with the physical world of humans and real life problem solving.

CO5: Design and implement IOT applications that manage big data

B.E., MECHANICAL ENGINEERING

ME3393 MANUFACTURING PROCESSES

COURSE OBJECTIVES:

1. To illustrate the working principles of various metal casting processes.
2. To learn and apply the working principles of various metal joining processes.
3. To analyse the working principles of bulk deformation of metals.
4. To learn the working principles of sheet metal forming process.
5. To study and practice the working principles of plastics molding.

UNIT – I METAL CASTING PROCESSES 9

Sand Casting – Sand Mould – Type of patterns - Pattern Materials – Pattern allowances – Molding sand Properties and testing – Cores –Types and applications – Molding machines – Types and applications– Melting furnaces – Principle of special casting processes- Shell, investment – Ceramic mould – Pressure die casting – low pressure, gravity- Tilt pouring, high pressure die casting- Centrifugal Casting – CO2 casting –-Defects in Sand casting process-remedies

UNIT II METAL JOINING PROCESSES 9

Fusion welding processes – Oxy fuel welding – Filler and Flux materials-–Arc welding, Electrodes, Coating and specifications – Gas Tungsten arc welding –Gas metal arc welding - Submerged arc welding – Electro slag welding– Plasma arc welding — Resistance welding Processes -Electron beam welding –Laser beam Welding Friction welding – Friction stir welding – Diffusion welding – Thermit Welding, Weld defects –inspection &remedies – Brazing - soldering – Adhesive bonding.

UNIT III BULK DEFORMATION PROCESSES 9

Hot working and cold working of metals – Forging processes – Open, impression and closed die forging – cold forging- Characteristics of the processes – Typical forging operations – rolling of metals – Types of Rolling – Flat strip rolling – shape rolling operations – Defects in rolled parts – Principle of rod and wire drawing – Tube drawing – Principles of Extrusion – Types – Hot and Cold extrusion. Introduction to shaping operations.

UNIT IV SHEET METAL PROCESSES 9

Sheet metal characteristics – Typical shearing, bending and drawing operations – Stretch forming operations – Formability of sheet metal – Test methods –special forminG processes - Working principle and applications – Hydro forming – Rubber pad forming Metal spinning – Introduction of Explosive forming, magnetic pulse forming, peen forming, Super plastic forming – Micro forming – Incremental forming.

UNIT V MANUFACTURE OF PLASTIC COMPONENTS 9

Types and characteristics of plastics – Molding of thermoplastics & Thermosetting polymers– working principles and typical applications – injection molding – Plunger and screw machines – Compression molding, Transfer Molding – Typical industrial applications – introduction to blow molding – Rotational molding – Film blowing - Extrusion – Thermoforming – Bonding of Thermoplastics- duff moulding.

COURSE OUTCOMES:

At the end of the course the students would be able to

1. Explain the principle of different metal casting processes.
2. Describe the various metal joining processes.
3. Illustrate the different bulk deformation processes.
4. Apply the various sheet metal forming process.
5. Apply suitable molding technique for manufacturing of plastics components.

DEPARTMENT OF INFORMATION TECHNOLOGY

CS3491 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

L	T	P	C
3	0	2	4

COURSE OBJECTIVES:

The main objectives of this course are to:

- Study about uninformed and Heuristic search techniques.
- Learn techniques for reasoning under uncertainty
- Introduce Machine Learning and supervised learning algorithms
- Study about ensembling and unsupervised learning algorithms
- Learn the basics of deep learning using neural networks

UNIT I PROBLEM SOLVING **9**

Introduction to AI - AI Applications - Problem solving agents – search algorithms – uninformed search strategies – Heuristic search strategies – Local search and optimization problems – adversarial search – constraint satisfaction problems (CSP)

UNIT II PROBABILISTIC REASONING **9**

Acting under uncertainty – Bayesian inference – naïve bayes models. Probabilistic reasoning – Bayesian networks – exact inference in BN – approximate inference in BN – causal networks.

UNIT III SUPERVISED LEARNING **9**

Introduction to machine learning – Linear Regression Models: Least squares, single & multiple variables, Bayesian linear regression, gradient descent, Linear Classification Models: Discriminant function – Probabilistic discriminative model - Logistic regression, Probabilistic generative model – Naive Bayes, Maximum margin classifier – Support vector machine, Decision Tree, Random forests

UNIT IV ENSEMBLE TECHNIQUES AND UNSUPERVISED LEARNING **9**

Combining multiple learners: Model combination schemes, Voting, Ensemble Learning – bagging, boosting, stacking, Unsupervised learning: K-means, Instance Based Learning: KNN, Gaussian mixture models and Expectation maximization

UNIT V NEURAL NETWORKS **9**

Perceptron - Multilayer perceptron, activation functions, network training – gradient descent optimization – stochastic gradient descent, error backpropagation, from shallow networks to deep networks –Unit saturation (aka the vanishing gradient problem) – ReLU, hyperparameter tuning, batch normalization, regularization, dropout.

45 PERIODS

PRACTICAL EXERCISES: **30 PERIODS**

1. Implementation of Uninformed search algorithms (BFS, DFS)
2. Implementation of Informed search algorithms (A*, memory-bounded A*)
3. Implement naïve Bayes models
4. Implement Bayesian Networks
5. Build Regression models
6. Build decision trees and random forests
7. Build SVM models
8. Implement ensembling techniques
9. Implement clustering algorithms
10. Implement EM for Bayesian networks
11. Build simple NN models
12. Build deep learning NN models

COURSE OUTCOMES:

At the end of this course, the students will be able to:

CO1: Use appropriate search algorithms for problem solving
CO2: Apply reasoning under uncertainty
CO3: Build supervised learning models
CO4: Build ensembling and unsupervised models
CO5: Build deep learning neural network models

TOTAL: 75 PERIODS

TEXT BOOKS:

1. Stuart Russell and Peter Norvig, “Artificial Intelligence – A Modern Approach”, FourthEdition, Pearson Education, 2021.
2. Ethem Alpaydin, “Introduction to Machine Learning”, MIT Press, Fourth Edition, 2020.

REFERENCES

1. Harry R Lewis and Christos H Papadimitriou , "Elements of the Theory of Computation", 2ndEdition, Prentice Hall of India, 2015.
2. Peter Linz, "An Introduction to Formal Language and Automata", 6th Edition, Jones & Bartlett,2016.
3. K.L.P.Mishra and N.Chandrasekaran, “Theory of Computer Science: Automata Languages andComputation”, 3rd Edition, Prentice Hall of India, 2006

IT3501 FULL STACK WEB DEVELOPMENT LTPC
3 3 0 3

COURSE OBJECTIVES:

- To understand the various components of full stack development
- To learn Node.js features and applications
- To develop applications with MongoDB
- To understand the role of Angular and Express in web applications
- To develop simple web applications with React

UNIT I BASICS OF FULL STACK 9

Understanding the Basic Web Development Framework - User - Browser – Webserver - Backend Services – MVC Architecture - Understanding the different stacks –The role of Express – Angular-Node – Mongo DB – React

UNIT II NODE JS 9

Basics of Node JS – Installation – Working with Node packages – Using Node package manager –Creating a simple Node.js application – Using Events – Listeners –Timers - Callbacks – Handling Data I/O – Implementing HTTP services in Node.js

UNIT III MONGO DB 9

Understanding NoSQL and MongoDB – Building MongoDB Environment – User accounts – Access control – Administering databases – Managing collections – Connecting to MongoDB from Node.js – simple applications

UNIT IV EXPRESS AND ANGULAR 9

Implementing Express in Node.js - Configuring routes - Using Request and Response objects - Angular - Typescript - Angular Components - Expressions - Data binding - Built-in directives

UNIT V REACT 9

MERN STACK – Basic React applications – React Components – React State – Express REST APIs - Modularization and Webpack - Routing with React Router – Server-side rendering

COURSE OUTCOMES:

At the end of the course, students will be able to

CO1: Understand the various stacks available for web application development

CO2: Use Node.js for application development

CO3: Develop applications with MongoDB

CO4: Use the features of Angular and Express

CO5: Develop React applications

TOTAL:45 PERIODS

TEXT BOOKS

1. Brad Dayley, Brendan Dayley, Caleb Dayley, 'Node.js, MongoDB and Angular WebDevelopment', Addison-Wesley, Second Edition, 2018
2. Vasan Subramanian, 'Pro MERN Stack, Full Stack Web App Development with Mongo,Express, React, and Node', Second Edition, Apress, 2019.

REFERENCES

1. Chris Northwood, 'The Full Stack Developer: Your Essential Guide to the Everyday SkillsExpected of a Modern Full Stack Web Developer', Apress; 1st edition, 2018
2. Kirupa Chinnathambi, 'Learning React: A Hands-On Guide to Building Web Applications Using React and Redux', Addison-Wesley Professional, 2nd edition, 2018
3. https://www.tutorialspoint.com/the_full_stack_web_development/index.asp
4. https://www.coursera.org/specializations/full-stack-react

https://www.udemy.com/course/the-full-stack-web-development/

CCS356 OBJECT ORIENTED SOFTWARE ENGINEERING L T P C
3 0 2 4

COURSE OBJECTIVES:

- To understand Software Engineering Lifecycle Models
- To Perform software requirements analysis
- To gain knowledge of the System Analysis and Design concepts using UML.
- To understand software testing and maintenance approaches
- To work on project management scheduling using DevOps

UNIT I SOFTWARE PROCESS AND AGILE DEVELOPMENT 9

Introduction to Software Engineering, Software Process, Perspective and Specialized Process Models –Introduction to Agility-Agile process-Extreme programming-XP Process-Case Study.

UNIT II REQUIREMENTS ANALYSIS AND SPECIFICATION 9

Requirement analysis and specification – Requirements gathering and analysis – Software Requirement Specification – Formal system specification – Finite State Machines – Petrinets –Object modelling using UML – Use case Model – Class diagrams – Interaction diagrams – Activity diagrams – State chart diagrams – Functional modelling – Data Flow Diagram- CASE TOOLS.

UNIT III SOFTWARE DESIGN 9

Software design – Design process – Design concepts – Coupling – Cohesion – Functional independence – Design patterns – Model-view-controller – Publish-subscribe – Adapter – Command – Strategy – Observer – Proxy – Facade – Architectural styles – Layered - Client Server - Tiered - Pipe and filter- User interface design-Case Study.

UNIT IV SOFTWARE TESTING AND MAINTENANCE **9**

Testing – Unit testing – Black box testing– White box testing – Integration and System testing– Regression testing – Debugging - Program analysis – Symbolic execution – Model Checking- Case Study

UNIT V PROJECT MANAGEMENT **9**

Software Project Management- Software Configuration Management - Project Scheduling- DevOps: Motivation-Cloud as a platform-Operations- Deployment Pipeline:Overall Architecture Building and Testing-Deployment- Tools- Case Study

COURSE OUTCOMES:

CO1: Compare various Software Development Lifecycle Models
CO2: Evaluate project management approaches as well as cost and schedule estimationstrategies.
CO3: Perform formal analysis on specifications.
CO4: Use UML diagrams for analysis and design.
CO5: Architect and design using architectural styles and design patterns, and test the system

45 PERIODS

PRACTICAL EXERCISES: **30 PERIODS**

LIST OF EXPERIMENTS:

1. Identify a software system that needs to be developed.
2. Document the Software Requirements Specification (SRS) for the identified system.
3. Identify use cases and develop the Use Case model.
4. Identify the conceptual classes and develop a Domain Model and also derive a Class Diagramfrom that.
5. Using the identified scenarios, find the interaction between objects and represent them usingUML Sequence and Collaboration Diagrams
6. Draw relevant State Chart and Activity Diagrams for the same system.
7. Implement the system as per the detailed design
8. Test the software system for all the scenarios identified as per the usecase diagram
9. Improve the reusability and maintainability of the software system by applying appropriatedesign patterns.
10. Implement the modified system and test it for various scenarios.

SUGGESTED DOMAINS FOR MINI-PROJECT:

1. Passport automation system.
2. Book bank
3. Exam registration
4. Stock maintenance system.
5. Online course reservation system
6. Airline/Railway reservation system

7. Software personnel management system
8. Credit card processing
9. e-book management system
10. Recruitment system
11. Foreign trading system
12. Conference management system
13. BPO management system
14. Library management system
15. Student information system

TOTAL:75 PERIODS

TEXT BOOKS

1. Bernd Bruegge and Allen H. Dutoit, "Object-Oriented Software Engineering: Using UML,Patterns and Java", Third Edition, Pearson Education, 2009.
2. Roger S. Pressman, Object-Oriented Software Engineering: An Agile Unified Methodology,First Edition, Mc Graw-Hill International Edition, 2014.

REFERENCES

1. Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, Fundamentals of Software Engineering, 2ndedition, PHI Learning Pvt. Ltd., 2010.
2. Craig Larman, Applying UML and Patterns, 3rd ed, Pearson Education, 2005.
3. Len Bass, Ingo Weber and Liming Zhu, —DevOps: A Software Architect's Perspective‖,Pearson Education, 2016
4. Rajib Mall, Fundamentals of Software Engineering, 3rd edition, PHI Learning Pvt. Ltd.,2009.
5. Stephen Schach, Object-Oriented and Classical Software Engineering, 8th ed, McGraw-Hill, 2010.

CCS370 UI AND UX DESIGN L T P C

2 0 2 3

COURSE OBJECTIVES:

- To provide a sound knowledge in UI & UX
- To understand the need for UI and UX
- To understand the various Research Methods used in Design
- To explore the various Tools used in UI & UX
- Creating a wireframe and prototype

UNIT I FOUNDATIONS OF DESIGN 6

UI vs. UX Design - Core Stages of Design Thinking - Divergent and Convergent Thinking - Brainstorming and Game storming - Observational Empathy

UNIT II FOUNDATIONS OF UI DESIGN 6

Visual and UI Principles - UI Elements and Patterns - Interaction Behaviors and Principles – Branding - Style Guides

UNIT III FOUNDATIONS OF UX DESIGN **6**

Introduction to User Experience - Why You Should Care about User Experience - Understanding User Experience - Defining the UX Design Process and its Methodology - Research in User Experience Design - Tools and Method used for Research - User Needs and its Goals - Know about Business Goals

UNIT IV WIREFRAMING, PROTOTYPING AND TESTING **6**

Sketching Principles - Sketching Red Routes - Responsive Design – Wireframing - Creating Wireflows - Building a Prototype - Building High-Fidelity Mockups - Designing Efficiently with Tools
- Interaction Patterns - Conducting Usability Tests - Other Evaluative User Research Methods - Synthesizing Test Findings - Prototype Iteration

UNIT V RESEARCH, DESIGNING, IDEATING, & INFORMATION ARCHITECTURE **6**

Identifying and Writing Problem Statements - Identifying Appropriate Research Methods - Creating Personas - Solution Ideation - Creating User Stories - Creating Scenarios - Flow Diagrams - Flow Mapping - Information Architecture

30 PERIODS

LIST OF EXPERIMENTS **30 PERIODS**

1. Designing a Responsive layout for an societal application
2. Exploring various UI Interaction Patterns
3. Developing an interface with proper UI Style Guides
4. Developing Wireflow diagram for application using open source software
5. Exploring various open source collaborative interface Platform
6. Hands on Design Thinking Process for a new product
7. Brainstorming feature for proposed product
8. Defining the Look and Feel of the new Project
9. Create a Sample Pattern Library for that product (Mood board, Fonts, Colors based on UIprinciples)
10. Identify a customer problem to solve
11. Conduct end-to-end user research - User research, creating personas, Ideation process (Userstories, Scenarios), Flow diagrams, Flow Mapping
12. Sketch, design with popular tool and build a prototype and perform usability testing andidentify improvements

TOTAL: 60 PERIODS

COURSE OUTCOMES:

On completion of the course, the students will be able to:

CO1: Build UI for user Applications

CO2: Evaluate UX design of any product or application

CO3: Demonstrate UX Skills in product development

CO4: Implement Sketching principles

CO5:Create Wireframe and Prototype

TEXT BOOKS

1. Joel Marsh, "UX for Beginners", O'Reilly , 2022
2. Jon Yablonski, "Laws of UX using Psychology to Design Better Product & Services" O'Reilly2021

REFERENCES

1. Jenifer Tidwell, Charles Brewer, Aynne Valencia, "Designing Interface" 3 rd Edition ,O'Reilly 2020
2. Steve Schoger, Adam Wathan "Refactoring UI", 2018
3. Steve Krug, "Don't Make Me Think, Revisited: A Commonsense Approach to Web &Mobile", Third Edition, 2015
4. https://www.nngroup.com/articles/
5. https://www.interaction-design.org/literature.

CCS344 ETHICAL HACKING **L T P C**
2 0 2 3

COURSE OBJECTIVES:

- To understand the basics of computer based vulnerabilities.
- To explore different foot printing, reconnaissance and scanning methods.
- To expose the enumeration and vulnerability analysis methods.
- To understand hacking options available in Web and wireless applications.
- To explore the options for network protection.
- To practice tools to perform ethical hacking to expose the vulnerabilities.

UNIT I INTRODUCTION **6**

Ethical Hacking Overview - Role of Security and Penetration Testers .- Penetration-Testing Methodologies- Laws of the Land - Overview of TCP/IP- The Application Layer - The Transport Layer - The Internet Layer - IP Addressing .- Network and Computer Attacks - Malware - Protecting Against Malware Attacks.- Intruder Attacks - Addressing Physical Security

UNIT II FOOT PRINTING, RECONNAISSANCE AND SCANNING NETWORKS 6

Footprinting Concepts - Footprinting through Search Engines, Web Services, Social Networking Sites, Website, Email - Competitive Intelligence - Footprinting through Social Engineering - Footprinting Tools - Network Scanning Concepts - Port-Scanning Tools - Scanning Techniques - Scanning Beyond IDS and Firewall

UNIT III ENUMERATION AND VULNERABILITY ANALYSIS **6**

Enumeration Concepts - NetBIOS Enumeration – SNMP, LDAP, NTP, SMTP and DNS Enumeration - Vulnerability Assessment Concepts - Desktop and Server OS Vulnerabilities - Windows OS Vulnerabilities - Tools for Identifying Vulnerabilities in Windows- Linux OS Vulnerabilities- Vulnerabilities of Embedded Oss

UNIT IV SYSTEM HACKING **6**

Hacking Web Servers - Web Application Components- Vulnerabilities - Tools for Web Attackers and Security Testers Hacking Wireless Networks - Components of a Wireless Network – Wardriving- Wireless Hacking - Tools of the Trade -

UNIT V NETWORK PROTECTION SYSTEMS **6**

Access Control Lists. - Cisco Adaptive Security Appliance Firewall - Configuration and Risk Analysis Tools for Firewalls and Routers - Intrusion Detection and Prevention Systems - Network- Based and Host-Based IDSs and IPSs - Web Filtering - Security Incident Response Teams – Honeypots.

30 PERIODS

PRACTICAL EXERCISES: **30 PERIODS**

1. Install Kali or Backtrack Linux / Metasploitable/ Windows XPPractice the
2. basics of reconnaissance.
3. Using FOCA / SearchDiggity tools, extract metadata and expanding the target list.
4. Aggregates information from public databases using online free tools likePaterva's Maltego.
5. Information gathering using tools like Robtex.
6. Scan the target using tools like Nessus.
7. View and capture network traffic using Wireshark.
8. Automate dig for vulnerabilities and match exploits using Armitage

FOCA : http://www.informatica64.com/foca.aspx.
Nessus : http://www.tenable.com/products/nessus.
Wireshark : http://www.wireshark.org.
Armitage : http://www.fastandeasyhacking.com/.
Kali or Backtrack Linux, Metasploitable, Windows XP

COURSE OUTCOMES:

At the end of this course, the students will be able:

CO1: To express knowledge on basics of computer based vulnerabilities
CO2: To gain understanding on different foot printing, reconnaissance and scanning methods.
CO3 To demonstrate the enumeration and vulnerability analysis methods
CO4: To gain knowledge on hacking options available in Web and wireless applications.
CO5: To acquire knowledge on the options for network protection.
CO6: To use tools to perform ethical hacking to expose the vulnerabilities.

TOTAL:60 PERIODS

TEXTBOOKS

1. Michael T. Simpson, Kent Backman, and James E. Corley, Hands-On Ethical Hacking and Network Defense, Course Technology, Delmar Cengage Learning, 2010.
2. The Basics of Hacking and Penetration Testing - Patrick Engebretson, SYNGRESS, Elsevier, 2013.
3. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, DafyddStuttard and Marcus Pinto, 2011.

REFERENCES

1. Black Hat Python: Python Programming for Hackers and Pentesters, Justin Seitz , 2014.

CCS333 AUGMENTED REALITY/VIRTUAL REALITY L T P C 2 0 2 3

COURSE OBJECTIVES:

- To impart the fundamental aspects and principles of AR/VR technologies.
- To know the internals of the hardware and software components involved in thedevelopment of AR/VR enabled applications.
- To learn about the graphical processing units and their architectures.
- To gain knowledge about AR/VR application development.
- To know the technologies involved in the development of AR/VR based applications.

UNIT I INTRODUCTION 7

Introduction to Virtual Reality and Augmented Reality – Definition – Introduction to Trajectories and Hybrid Space-Three I's of Virtual Reality – Virtual Reality Vs 3D Computer Graphics – Benefits of Virtual Reality – Components of VR System – Introduction to AR-AR Technologies- Input Devices – 3D Position Trackers – Types of Trackers – Navigation and Manipulation Interfaces – Gesture Interfaces – Types of Gesture Input Devices – Output Devices – Graphics Display – Human Visual System – Personal Graphics Displays – Large Volume Displays – Sound Displays – Human Auditory System.

UNIT II VR MODELING 6

Modeling – Geometric Modeling – Virtual Object Shape – Object Visual Appearance – Kinematics Modeling – Transformation Matrices – Object Position – Transformation Invariants –Object Hierarchies – Viewing the 3D World – Physical Modeling – Collision Detection – Surface Deformation – Force Computation – Force Smoothing and Mapping – Behavior Modeling – Model Management.

UNIT III VR PROGRAMMING 6

VR Programming – Toolkits and Scene Graphs – World ToolKit – Java 3D –

Comparison of World ToolKit and Java 3D

UNIT IV APPLICATIONS **6**

Human Factors in VR – Methodology and Terminology – VR Health and Safety Issues – VR and Society-Medical Applications of VR – Education, Arts and Entertainment – Military VR Applications

– Emerging Applications of VR – VR Applications in Manufacturing – Applications of VR in Robotics – Information Visualization – VR in Business – VR in Entertainment – VR in Education.

UNIT V AUGMENTED REALITY **5**

Introduction to Augmented Reality-Computer vision for AR-Interaction-Modelling and Annotation- Navigation-Wearable devices

30 PERIODS

PRACTICAL EXERCISES:

1. Study of tools like Unity, Maya, 3DS MAX, AR toolkit, Vuforia and Blender.
2. Use the primitive objects and apply various projection types by handling camera.
3. Download objects from asset store and apply various lighting and shading effects.
4. Model three dimensional objects using various modelling techniques and apply texturesover them.
5. Create three dimensional realistic scenes and develop simple virtual reality enabled mobileapplications which have limited interactivity.
6. Add audio and text special effects to the developed application.
7. Develop VR enabled applications using motion trackers and sensors incorporating fullhaptic interactivity.
8. Develop AR enabled applications with interactivity like E learning environment, Virtualwalkthroughs and visualization of historic places.
9. Develop AR enabled simple applications like human anatomy visualization, DNA/RNAstructure visualization and surgery simulation.
10. Develop simple MR enabled gaming applications.

30 PERIODS

COURSE OUTCOMES:

On completion of the course, the students will be able to:

CO1: Understand the basic concepts of AR and VR

CO2: Understand the tools and technologies related to AR/VR

CO3: Know the working principle of AR/VR related Sensor devices

CO4: Design of various models using modeling techniques

CO5: Develop AR/VR applications in different domains

TOTAL:60 PERIODS

TEXTBOOKS:

1. Charles Palmer, John Williamson, “Virtual Reality Blueprints: Create compelling

VRexperiences for mobile", Packt Publisher, 2018
2. Dieter Schmalstieg, Tobias Hollerer, "Augmented Reality: Principles & Practice", AddisonWesley, 2016
3. John Vince, "Introduction to Virtual Reality", Springer-Verlag, 2004.
4. William R. Sherman, Alan B. Craig: Understanding Virtual Reality – Interface, Application,Design", Morgan Kaufmann, 2003

GE3151 PROBLEM SOLVING AND PYTHON PROGRAMMING **L T P C**
3 0 0 3

COURSE OBJECTIVES:

- To understand the basics of algorithmic problem solving.
- To learn to solve problems using Python conditionals and loops.
- To define Python functions and use function calls to solve problems.
- To use Python data structures - lists, tuples, dictionaries to represent complex data.
- To do input/output with files in Python.

UNIT I COMPUTATIONAL THINKING AND PROBLEM SOLVING **9**

Fundamentals of Computing – Identification of Computational Problems -Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi.

UNIT II DATA TYPES, EXPRESSIONS, STATEMENTS **9**

Python interpreter and interactive mode,debugging; values and types: int, float, boolean, string , and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.

UNIT III CONTROL FLOW, FUNCTIONS, STRINGS **9**

Conditionals:Boolean values and operators, conditional (if), alternative (if-else),chained conditional (if-elif-else);Iteration: state, while, for, break, continue, pass; Fruitful functions: return values,parameters, local and global scope, function composition, recursion; Strings: string slices,immutability, string functions and methods, string module; Lists as arrays. Illustrative programs: square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.

UNIT IV LISTS, TUPLES, DICTIONARIES **9**

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension; Illustrative programs: simple sorting, histogram, Students marks statement, Retail bill preparation.

UNIT V FILES, MODULES, PACKAGES **9**

Files and exceptions: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file, Voter's age validation, Marks range validation (0-100).

TOTAL : 45 PERIODS

COURSE OUTCOMES:

Upon completion of the course, students will be able to

CO1: Develop algorithmic solutions to simple computational problems.
CO2: Develop and execute simple Python programs.
CO3: Write simple Python programs using conditionals and loops for solving problems.
CO4: Decompose a Python program into functions.
CO5: Represent compound data using Python lists, tuples, dictionaries etc.
CO6: Read and write data from/to files in Python programs.

TEXT BOOKS:

1. Allen B. Downey, "Think Python: How to Think like a Computer Scientist", 2nd Edition, O'Reilly Publishers, 2016.
2. Karl Beecher, "Computational Thinking: A Beginner's Guide to Problem Solving and Programming", 1st Edition, BCS Learning & Development Limited, 2017.

REFERENCES:

1. Paul Deitel and Harvey Deitel, "Python for Programmers", Pearson Education, 1st Edition, 2021.
2. G Venkatesh and Madhavan Mukund, "Computational Thinking: A Primer for Programmersand Data Scientists", 1st Edition, Notion Press, 2021.
3. John V Guttag, "Introduction to Computation and Programming Using Python: With Applications to Computational Modeling and Understanding Data", Third Edition, MIT Press, 2021
4. Eric Matthes, "Python Crash Course, A Hands - on Project Based Introduction to Programming", 2nd Edition, No Starch Press, 2019.
5. https://www.python.org/
6. Martin C. Brown, "Python: The Complete Reference", 4th Edition, Mc-Graw Hill, 2018.

CCS342 DEVOPS L T P C2 0 2 3

COURSE OBJECTIVES:

- To introduce DevOps terminology, definition & concepts
- To understand the different Version control tools like Git, Mercurial
- To understand the concepts of Continuous Integration/ Continuous Testing/ ContinuousDeployment)
- To understand Configuration management using Ansible
- Illustrate the benefits and drive the adoption of cloud-based Devops tools to solve realworld problems

UNIT I INTRODUCTION TO DEVOPS **6**

Devops Essentials - Introduction To AWS, GCP, Azure - Version control systems: Git and Github.

UNIT II COMPILE AND BUILD USING MAVEN & GRADLE 6

Introduction, Installation of Maven, POM files, Maven Build lifecycle, Build phases(compile build, test, package) Maven Profiles, Maven repositories(local, central, global),Maven plugins, Maven create and build Artificats, Dependency management, Installation of Gradle, Understand build using Gradle

UNIT III CONTINUOUS INTEGRATION USING JENKINS 6

Install & Configure Jenkins, Jenkins Architecture Overview, Creating a Jenkins Job, Configuring a Jenkins job, Introduction to Plugins, Adding Plugins to Jenkins, Commonly used plugins (Git Plugin, Parameter Plugin, HTML Publisher, Copy Artifact and Extended choice parameters). Configuring Jenkins to work with java, Git and Maven, Creating a Jenkins Build and Jenkins workspace.

UNIT IV CONFIGURATION MANAGEMENT USING ANSIBLE 6

Ansible Introduction, Installation, Ansible master/slave configuration, YAML basics, Ansible modules, Ansible Inventory files, Ansible playbooks, Ansible Roles, adhoc commands in ansible

UNIT V BUILDING DEVOPS PIPELINES USING AZURE 6

Create Github Account, Create Repository, Create Azure Organization, Create a new pipeline,Build a sample code, Modify azure-pipelines.yaml file

COURSE OUTCOMES:

CO1: Understand different actions performed through Version control tools like Git.

CO2: Perform Continuous Integration and Continuous Testing and Continuous Deploymentusing Jenkins by building and automating test cases using Maven & Gradle.

CO3: Ability to Perform Automated Continuous Deployment

CO4: Ability to do configuration management using Ansible

CO5: Understand to leverage Cloud-based DevOps tools using Azure DevOps

30 PERIODS

PRACTICAL EXERCISES: 30 PERIODS

1. Create Maven Build pipeline in Azure
2. Run regression tests using Maven Build pipeline in Azure
3. Install Jenkins in Cloud
4. Create CI pipeline using Jenkins
5. Create a CD pipeline in Jenkins and deploy in Cloud
6. Create an Ansible playbook for a simple web application infrastructure
7. Build a simple application using Gradle
8. Install Ansible and configure ansible roles and to write playbooks

TOTAL:60 PERIODS

TEXT BOOKS

1. Roberto Vormittag, "A Practical Guide to Git and GitHub for Windows Users: From Beginnerto Expert in Easy Step-By-Step Exercises", Second Edition, Kindle Edition, 2016.
2. Jason Cannon, "Linux for Beginners: An Introduction to the Linux Operating System and Command Line", Kindle Edition, 2014

REFERENCES

1. Hands-On Azure Devops: Cicd Implementation For Mobile, Hybrid, And Web Applications Using Azure Devops And Microsoft Azure: CICD Implementation for ... DevOps and Microsoft Azure (English Edition) Paperback – 1 January 2020
2. by Mitesh Soni
3. Jeff Geerling, "Ansible for DevOps: Server and configuration management for humans", First Edition, 2015.
4. David Johnson, "Ansible for DevOps: Everything You Need to Know to Use Ansible for DevOps", Second Edition, 2016.
5. Mariot Tsitoara, "Ansible 6. Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer", Second Edition, 2019.
6. https://www.jenkins.io/user-handbook.pdf
7. https://maven.apache.org/guides/getting-started/

CCS335 CLOUD COMPUTING L T P C2 0 2 3

COURSE OBJECTIVES:

- To understand the principles of cloud architecture, models and infrastructure.
- To understand the concepts of virtualization and virtual machines.
- To gain knowledge about virtualization Infrastructure.
- To explore and experiment with various Cloud deployment environments.
- To learn about the security issues in the cloud environment.

UNIT I CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE 6

Cloud Architecture: System Models for Distributed and Cloud Computing – NIST Cloud ComputingReference Architecture – Cloud deployment models – Cloud service models; Cloud Infrastructure: Architectural Design of Compute and Storage Clouds – Design Challenges

UNIT II VIRTUALIZATION BASICS 6

Virtual Machine Basics – Taxonomy of Virtual Machines – Hypervisor – Key Concepts – Virtualization structure – Implementation levels of virtualization – Virtualization Types: Full Virtualization – Para Virtualization – Hardware Virtualization – Virtualization of CPU, Memory and I/O devices.

UNIT III VIRTUALIZATION INFRASTRUCTURE AND DOCKER 7

Desktop Virtualization – Network Virtualization – Storage Virtualization – System-level of Operating Virtualization – Application Virtualization – Virtual clusters and Resource

Management – Containers vs. Virtual Machines – Introduction to Docker – Docker Components – Docker Container – Docker Images and Repositories.

UNIT IV CLOUD DEPLOYMENT ENVIRONMENT **6**

Google App Engine – Amazon AWS – Microsoft Azure; Cloud Software Environments – Eucalyptus – OpenStack.

UNIT V CLOUD SECURITY **5**

Virtualization System-Specific Attacks: Guest hopping – VM migration attack – hyperjacking. Data Security and Storage; Identity and Access Management (IAM) - IAM Challenges - IAM Architectureand Practice.

30 PERIODS

PRACTICAL EXERCISES: **30 PERIODS**

1. Install Virtualbox/VMware/ Equivalent open source cloud Workstation with differentflavours of Linux or Windows OS on top of windows 8 and above.
2. Install a C compiler in the virtual machine created using a virtual box and execute SimplePrograms
3. Install Google App Engine. Create a hello world app and other simple web applicationsusing python/java.
4. Use the GAE launcher to launch the web applications.
5. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is notpresent in CloudSim.
6. Find a procedure to transfer the files from one virtual machine to another virtual machine.
7. Install Hadoop single node cluster and run simple applications like wordcount.
8. Creating and Executing Your First Container Using Docker.
9. Run a Container from Docker Hub

COURSE OUTCOMES:

CO1: Understand the design challenges in the cloud.
CO2: Apply the concept of virtualization and its types.
CO3: Experiment with virtualization of hardware resources and Docker.
CO4: Develop and deploy services on the cloud and set up a cloud environment.
CO5: Explain security challenges in the cloud environment.

TOTAL :60 PERIODS

TEXT BOOKS

1. Kai Hwang, Geoffrey C Fox, Jack G Dongarra, “Distributed and Cloud Computing, From Parallel Processing to the Internet of Things”, Morgan Kaufmann Publishers, 2012.
2. James Turnbull, “The Docker Book”, O’Reilly Publishers, 2014.
3. Krutz, R. L., Vines, R. D, “Cloud security. A Comprehensive Guide to Secure CloudComputing”, Wiley Publishing, 2010.

REFERENCES

1. James E. Smith, Ravi Nair, “Virtual Machines: Versatile Platforms for Systems

andProcesses", Elsevier/Morgan Kaufmann, 2005.

2. Tim Mather, Subra Kumaraswamy, and Shahed Latif, "Cloud Security and Privacy: an enterprise perspective on risks and compliance", O'Reilly Media, Inc., 2009.