BCA (Cloud Computing)

Semester-V

L-3 T-1 P-0 C-4

25BCC170T: Cloud Web Services

Course Objectives

- To learn Introduction about AWS and Identity access management policy.
- Learn AWS Storage and Compute services.
- To Learn about AWS Database and Networking.
- To get knowledge about different AWS Applications Services.
- To learn Cloud Server less Designing Principles.

Course Outcomes (COs)

1. Understand the fundamental concepts of AWS, including its history, global infrastructure, and IAM service.
2. Apply AWS storage and compute services such as S3, EC2, EBS, and EFS to build scalable and reliable cloud solutions.
3. Analyze AWS database services like RDS, DynamoDB, and networking services for secure and efficient data management.
4. Evaluate AWS application services such as SQS, SNS, and API Gateway for building scalable and decoupled architectures.
5. Create serverless architectures using AWS Lambda, manage costs effectively, and set up billing alarms to optimize resource usage.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	-	3	1	-	2	-	-	-	-	-	-	-
CO2	1	-	3	-	-	2	-	-	2	-	-	-
CO3	2	1	-	3	-	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	1	-	-	2	1
CO5	-	1	2	-	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit-I: Introduction & AWS Overview **12 Hours**

Introduction & AWS Overview: Introduction Cloud Computing, The History of AWS, AWS overview, AWS Global Infrastructure, AWS free tier sign up, IAM service Overview, Policies, Group, Users Management.

Unit-II: AWS Storage Services **12 Hours**

Storage & Compute Service:S3 Introduction, Create an S3 Buckets, S3 Life cycle management and glacier, Snowball, Snow edge, Snowmobile.EC2 Introduction, EC2 Linux Instance, EC2 Windows Instance, Security group, EBS, Volumes and Snapshots, Elastic File

System, S3 Glacier.

Unit-III: AWS Database and Networking Services: **12 Hours**
Database on AWS & Networking Services: Database Introduction, Create a RDS Instance, DynamoDB, Aurora, Redshift, Elasticache. DNS introduction, Route S3, Register a Domain Name. Introduction VPC, Build a Custom VPC, NAT, ACL, Firewall.

Unit-IV: Applications Services: **12 Hours**
Applications Services: Simple Queue Service, Simple Notification Service, Simple Workflow Service, Elastic Transcoder, API gateway.

Unit-V: AWS Serverless Architecture & AWS Costs **12 Hours**
AWS Serverless Architecture & AWS Costs: Introduction about Lambda,learn about AWS pricing, Optimize your Cost. Create Billing Alarm.

Total: 60 Hours

Reference Books:

1. AWS Documentation.
2. Amazon Web Services for Dummies, Bernald Golden, John Wiley & Sons, 2013 Reference Book.
3. Amazon Web Services Tutorial for Beginners: Bert David.
4. Brief Guide to Cloud Computing, Christopher Barnett, Constable & Robinson Limited, 2010.
5. "AWS Certified Solutions Architect - Official Study Guide: Associate Exam", Joe Baron, Hisham Baz, Tim Bixler, Biff Gaut, Kevin E. Kelly, Sean Senior, John Stamper, Sybex.
6. "Amazon Web Services in Action"**,** Andreas Wittig and Michael Wittig, Manning Publications.
7. "AWS Certified Cloud Practitioner Study Guide: CLF-C01 Exam", Ben Piper and David Clinton, Wiley Publications.

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/
3. https://www.udemy.com/
4. https://www.edx.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

BCA (Cloud Computing)

Semester-V

L-3 T-0 P-0 C-3

25BCC180T: Cloud Security

Course Objectives

- To know about Network Security.
- To learn Cloud Computing Security Fundamentals.
- To assess risk tolerance, addressing security concerns and regulatory aspects.
- To utilize checklists and metrics to evaluate the security posture of cloud environments effectively.
- Developing and implementing robust security policies.

Course Outcomes

1. Understand and implement network security policies and cloud delivery models.
2. Apply fundamental cloud computing security principles and design secure cloud software requirements.
3. Analyze security concerns, risk assessments, and legal aspects related to cloud computing.
4. Evaluate cloud security through checklists and metrics for comprehensive assessment.
5. Address cloud computing security challenges with effective policy implementation and best practice techniques.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	-	3	1	-	-	2	-	1	-	-	-	-
CO2	-	-	3	1	2	-	-	-	2	-	-	-
CO3	-	1	2	3	-	-	1	-	-	1	-	-
CO4	1	2	-	-	3	-	-	-	-	-	2	1
CO5	-	-	1	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit-I: Network Security **9 Hours**

Network Security - Determination of the network security policies, implementation network security policies, Reconnaissance, Vulnerability scanning, Penetration testing, post attack investigation, Cloud Delivery Models: The SPI Framework, SPI Evolution, the SPI Framework vs. the Traditional IT Model.

Unit-II: Cloud Computing Security Fundamentals **9 Hours**

Cloud Computing Security Fundamentals - Cloud Information Security Objectives, Confidentiality, Integrity, and Availability, Security attacks and threats, Cloud Security Services, Relevant Cloud Security Design Principles, and Approaches to Cloud Software Requirements Engineering, Cloud Security Policy Implementation and Decomposition

Unit-III: Security Concerns, Risk Issues, and Legal Aspects 9 Hours
Security Concerns, Risk Issues, and Legal Aspects - Cloud Computing: Security Concerns, Assessing Your Risk Tolerance in Cloud Computing, Legal and Regulatory Issues.

Unit-IV: Evaluating Cloud Security 9 Hours
Evaluating Cloud Security - Evaluating Cloud Security, Checklists for Evaluating Cloud Security, Metrics for the Checklists

Unit-V: Cloud Computing Security Challenges 9 Hours
Cloud Computing Security Challenges - Security Policy Implementation, Policy Types, Senior Management Statement of Policy, Regulatory Policies ,Advisory Policies, Informative Policies, Computer Security Incident Response Team (CSIRT),Virtualization Security Management, Virtual Threats, Hypervisor Risks, Increased Denial of Service Risk, VM Security Recommendations, Best Practice Security Techniques, VM-Specific Security Techniques, Hardening the Virtual Machine Securing, VM Remote Access.

Total Hours: 45

Reference Books:

1. Securing the Cloud: Cloud Computer Security Techniques and Tactics by Vic (J.R.) Winkler, Syngress (1 June 2011)
2. Practical Cloud Security: A Cross-Industry View by Melvin B. Greer Jr., Kevin L. Jackson CRC Press; 1 edition (2 August 2016)
3. CCSP (ISC)2 Certified Cloud Security Professional Official Study Guide 1st , Kindle Edition by Ben Malisow.
4. Cloud Security – A comprehensive Guide to Secure Cloud Computing by Ronald L. Krutz And Russel Dean Vines.
5. OpenStack Cloud Security by Fabio Alessandro Locati, Packt Publishing Limited (28 July 2015)
6. Cloud Security – A comprehensive Guide to Secure Cloud Computing by Ronald L. Krutz and Russel Dean Vines, Wiley, 2010
7. Cloud Security and Privacy by Mather Tim, Shroff Publishers & Distributers Private Limited - Mumbai; First edition (2009)

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/
3. https://udemy.com/
4. https://www.edx.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

BCA(System Administration and Cyber Security)

Semester-V

L-3 T-1 P-0 C-4

25BCC191T: Programming in JAVA

Course Objectives

- To learn about the basic introduction to Java.
- To learn about classes, objects, and methods.
- To learn and practice the Collection framework.
- To know about packages.
- To know about event handling.

Course Outcomes

Students will be able to:

1. Understand the concepts of Java.
2. Apply the useful classes, objects, and methods.
3. Analyze the Collection framework.
4. Evaluate the strengths of packages.
5. Create a probabilistic event handling concept.

Articulation Matrix:-

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation))

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	-	3	1	-	2	-	-	-	-	-	-	-
CO2	-	1	3	2	-	-	-	-	2	-	-	-
CO3	-	2	-	3	-	-	1	-	-	-	-	-
CO4	1	2	-	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit-: Introduction to Java : **12 Hours**

Introduction to Java, History of Java: Comparison of Java and C++, Java as an object-oriented language: Java Features, JDK, JRE, Java virtual Machine, comments, Data types, Operators, Operators precedence and associativity, escape sequences, Type casting, Type Conversion The decision making–if, if.... else, switch, loops–for, while, do...while, special statements–return, break, continue, labeled break, labeled continue, arrays, keywords.

Unit-II: CLASSES, OBJECTS AND METHODS : **12 Hours**

Introduction; Defining a Class; Adding Variables; Adding Variables; Adding Methods; Creating Objects; Accessing Class Members; Constructors; Methods Overloading; Static Members; Nesting of Methods; Inheritance: Extending a Class; Overriding Methods; Final Variables and Methods; Final Classes; Finalizer Methods; Abstract Methods and Classes; Visibility Control, Math class, Random Class

Unit-III: Collection framework : 12 Hours
LinkedList –HashSet, TreeSet, Hashtable, Strings, String functions, ArrayList, Traversing an ArrayList: using for-each loop, Iterator, ListIterator, Wrapper Classes, Auto Boxing and Unboxing. INTERFACES: Introduction; Defining Interfaces; Extending Interfaces; Implementing Interfaces; Accessing Interface Variables, Implementing Multiple Inheritance using Interfaces.

Unit-IV: PACKAGES : 12 Hours
Introduction; System Packages; Using System Packages; Naming Conventions; Creating Packages; Accessing a Package; Using a Package; Adding a Class to a Package; Hiding Classes.
Multithreading and Exception Handling: Basic idea of multithreaded programming, the lifecycle of a thread, creating thread with the thread class and runnable interface, Basic idea of exception handling, the try, catch and throw, throws and finally in exception handling.

Unit-V: Event Handling: 12 Hours
Delegation Event Model, Events, Event classes, Event listener interfaces, Using delegation event model, adapter classes and inner classes. Abstract Window Toolkit: Window Fundamentals, Component, Container, Panel, Window, Frame, Canvas. Components – Labels, Buttons, Check Boxes, RadioButtons, Choice Menus, TextFields, Text, ScrollingList, Scrollbars, Panels, Frames. Layouts: Flow Layout, Grid Layout, Border Layout, Card Layout
Swings: Classes, Working with JFrame Windows, Working with Graphics, Working with Colour, Adding and Removing Controls, Responding to Controls, Labels, Buttons, Checkbox, Checkbox Group, Choice Control, Lists, Text Field, Text Area. Menus, Dialog Box, Handling Events.

Total: 60 Hours

Reference Books:

1. Naughton & Schildt "The Complete Reference Java 2", Tata McGraw Hill
2. D. S. Malik, "Java Programming: From Problem Analysis to Program Design",Cengage Learning
3. "Java: How to Program", Paul Deitel and Harvey Deitel, Pearson
4. "Java Swing", Marc Loy, Robert Eckstein, Dave Wood, and James Elliott, O'Reilly Media

List of e-Learning Resources:

1.https://nptel.ac.in/
2.https://www.coursera.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

BCA (Cloud Computing)

Semester-V

L-3 T-1 P-0 C-4

25BCC192T: Advanced Server-side Scripting Using Node.js

Course Objectives

- Learn basics of Node JS and to set up a development environment.
- To learn about Installing Packages and Adding dependency.
- To learn file handling and events.
- To learn database connectivity.
- To learn Node JS Template engine and RESTFul API

Course Outcomes

1. Understand Node.js basics, advantages, process model, and setting up a development environment on Windows.
2. Apply npm for package management, create web servers, and handle HTTP requests effectively in Node.js.
3. Analyze file system operations, debugging techniques, and event handling using Event Emitters in Node.js.
4. Implement Express.js for routing, serving static resources, and database connectivity in Node.js applications.
5. Create dynamic web applications using template engines, explore utility modules, and scale applications with Express.js and RESTful APIs.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	-	3	1	-	2	-	-	-	-	-	-	-
CO2	-	-	3	1	-	2	-	-	2	-	-	-
CO3	-1	-	2	3	-	-	1	-	-	-	-	-
CO4	2	-	1	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit-I: Introduction to Node JS **12 Hours**

Introduction, Advantages of Node JS, Traditional Web Server Model, Node.js Process Model, **Setup Dev Environment:** Install Node.js on Windows, Working in REPL terminal, Node JS Console, Functions, Buffer, Module, Module Types, Core Modules, Local Modules ,Module. Exports

Unit-II: Package Management **12 Hours**

Package Manager (NEPM), Installing Packages Locally, Adding dependency in package.json, installing packages globally, Updating packages, creating web server, Handling http requests, Sending requests

Unit-III: File Handling **12 Hours**

Fs.read File, Writing a File, Writing a file asynchronously, Opening a file, Deleting a file, Other IO Operations, Core Node JS debugger, Debugging with Visual Studio

Events: Event Emitter class, Returning event emitter, Inhering events.

Unit-IV: Express.JS **12 Hours**

Express.JS: Configuring routes, working with express. **Serving Static Resources:** Serving static files, working with middleware. **Database connectivity:** Connection string, Configuring Working with select command, updating records, Deleting records

Unit-V: Probabilistic Reasoning and Uncertainty **12 Hours**

Template Engine, What is Jade, What is vash, Global Objects, Utility Modules, Web Module, Express Framework, RESTFul API, Scaling Application, and Packaging.

Total Hours: 60

Reference Books:

1. Node.js IN ACTION: by “Mike Cantelon, Marc Harter T.J., Holowaychuk, Nathan Rajlic” “Manning Publications Co.”
2. Learning Node: Moving to the Server-Side by “Shelley Powers” O’REILLY
3. "Node.js in Action" by Mike Cantelon, Marc Harter, T.J. Holowaychuk, and Nathan Rajlich
4. "Learning Node.js: A Hands-On Guide to Building Web Applications in JavaScript" by Marc Wandschneider:
5. "Node.js 8 the Right Way: Practical, Server-Side JavaScript That Scales" by Jim Wilson

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/
3. https://udemy.com/
4. http://www.edx.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

BCA (Cloud Computing)

Semester-V

L-3 T-1 P-0 C-4

25BCC193T: Web Application Development Using Python

Course Objectives

- To know about basic concepts of Python.
- To learn about Django.
- To learn and practice Django URL Pattern.
- To know about Django Form and Field validation.
- To know about Authentication.

Course Outcomes

1. Understand the concepts of python
2. Apply the useful python programming with django
3. Analyze the Django URL Pattern
4. Evaluate the Django Form and Field validation
5. Create probabilistic reasoning project for Authentication and Advanced Forms processing techniques

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	2	3	1	-	-	-	-	-	-	-	-	-
CO2	-	2	3	1	-	-	-	-	2	-	-	-
CO3	-	1	2	3	-	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	1	1	2

High-3 Medium-2 Low-1

Unit-I **12 Hours**

Revision of Python basics: Data types, variables, conditional statements, loops etc, Introduction to various frameworks, introduction to DJango framework for web development, installing and configuring DJango and Python for web development.

Unit-II **12 Hours**

Getting started with Django: Setting up database connections, Managing Users & the Django admin tool, Installing and using ‘out of the box’ Django features

Unit-III **12 Hours**

Django URL Patterns and Views: Designing a good URL scheme, Generic Views, **MVT:** Django’s take on MVC: Model, View and Template, Small application using MVT, About the 3 Core Files: models.py, urls.py, views.py, Django project concepts

Unit-IV **12 Hours**

Django Form and Field validation: Form classes, Validation: is_valid, full_clean, validate (), to_python (), run_validators (), clean (), raising Validation_Error

Unit-V **12 Hours**

Authentication, Advanced Forms processing techniques, Managing Users & the Django admin tool, Installing and using 'out of the box' Django features, DJango Session, Deploying DJango, Cache framework, Security, Django admin site, Request Response object,Django template.

Total 60 Hours

Reference Books:

1. Django Unleashed: by Andrew Pinkham
2. The Definitive Guide to Django: by Adrian Holovaty
3. Lightweight Django by Elman and Lavin

List of e-Learning Resources:

1.https://nptel.ac.in/
2.https://www.coursera.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

BCA (Cloud Computing)

Semester-V

L-3 T-0 P-0 C-3

25BCC201T: Internet of Things

Course Objectives

- To understand the basic concepts and Characteristics of IOT.
- To learn Design Principles for Web Connectivity.
- To understand Sensor Technology.
- To learn IOT Design methodology.
- To learn to Develop IoT solutions.

Course Outcomes

1. Understand the foundational concepts of IoT.
2. Apply design principles for web connectivity and internet communication protocols.
3. Analyze sensor technologies, participatory sensing, and communication protocols for sensor data.
4. Implement IoT design methodologies, address privacy and security concerns.
5. Develop IoT solutions using tools like Arduino and Raspberry Pi.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	1	3	-	2	-	-	-	-	-	-	-	-
CO2	-	2	3	-	1	-	-	1	2	-	-	-
CO3	-	1	-	3	-	2	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	-	-	-	1	2
CO5	-	1	-	2	-	3	-	-	-	-	2	1

High-3 Medium-2 Low-1

Unit-I: Introduction to IoT 9 Hours

Introduction: Definition, Characteristics of IOT, IOT Conceptual framework, IOT Architectural view, Physical design of IOT, Logical design of IOT, Application of IOT. Machine-to-machine (M2M), SDN (software defined networking) and NFV(network function virtualization) for IOT, data storage in IOT, IOT Cloud Based Services.

Unit-II: Design Principles for Web Connectivity 9 Hours

Design Principles for Web Connectivity: Web Communication Protocols for connected devices, Message Communication Protocols for connected devices, SOAP, REST, HTTP Restful and Web Sockets. Internet Connectivity Principles: Internet Connectivity, Internet based communication, IP addressing in IOT, Media Access control.

Unit-III: Sensor Technology 9 Hours

Sensor Technology, Participatory Sensing, Industrial IOT and Automotive IOT , Actuator, Sensor data Communication Protocols ,Radio Frequency Identification Technology, Wireless Sensor Network Technology.

Unit-IV: IOT Design methodology **9 Hours**

IOT Design methodology: Specification -Requirement, process, model, service, functional & operational view.IOT Privacy and security solutions, Raspberry Pi &arduino devices. IOT Case studies: smart city streetlights control & monitoring.

Unit-V: Developing IoT solutions **9 Hours**

Developing IoT solutions: Introduction to different IoT tools, Introduction to Arduino and Raspberry Pi Implementation of IoT with Arduino and Raspberry, Cloud Computing, Fog Computing, Connected Vehicles, Data Aggregation for the IoT in Smart Cities, Privacy and Security Issues in IoT.

Total Hours: 45

List of Experiments

1. Introduction to various sensors and various actuators & its Application (Students have to prepare Report for the same). Perform Experiment using Arduino Uno to measure the distance of any object using Ultrasonic Sensor.
   a. PIR Motion Sensor.
   b. Rain Drop Sensor.
   c. Moisture Sensor.
   d. Temperature Sensor.
   e. Touch Sensor.
   f. Infrared Sensor.
   g. Servo Moto.
   h. RFID Sensor.
   i. Bluetooth Module.
   j. Wi-Fi Module.
2. Demonstrate NodeMCU and its working
3. Getting Started with ESP8266 Wi-Fi SoC
4. Hands-on with on-board peripherals of ESP8266
5. Demonstrate Arduino and its pins.
6. Perform Experiment using Arduino Uno to measure the distance of any object using Ultrasonic Sensor.
7. Create a circuit using Arduino and sensors. Perform experiment using Arduino Uno to Learn Working of Servo Motor
8. Creating a webpage and display the values available through Arduino.
9. Demonstration of Setup & Working of Raspberry Pi. (Students have to prepare the Report for the same.).
10. OPEN Ended problem: Students are required to submit an IOT based project using the Microcontroller or a Raspberry Pi and connecting various sensors and actuators. The data for the same should be displayed via a webpage or a web app.

Reference Book:

1. Rajkamal,"Internet of Things", Tata McGraw Hill publication
2. Vijay Madisetti and ArshdeepBahga, "Internet of things(A-Hand-on-Approach)" 1st Edition ,Universal Press
3. HakimaChaouchi "The Internet of Things: Connecting Objects", Wiley publication.
4. Charless Bell "MySQL for the Internet of things", Apress publications.

5. Francis dacosta “Rethinking the Internet of things:A scalable Approach to connecting everything”, 1st edition, Apress publications 2013.
6. Donald Norris“The Internet of Things: Do-It-Yourself at Home Projects for Arduino, Raspberry Pi and BeagleBone Black”, McGraw Hill publication.
7. Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, “From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence”, 1st Edition, Academic Press, 2014. Syllabus for Bachelor of Technology Computer Engineering
8. Cuno Pfister, Getting Started with the Internet of Things, O‟Reilly Media, 2011, ISBN: 978-1-4493- 9357-1

List of e-Learning Resources:

- https://nptel.ac.in/
- https://www.coursera.org/
- https://www.edx.org/
- https://udemy.com/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

BCA (Cloud Computing)

Semester-V

L-3 T-0 P-0 C-3

25BCC202T: Software Engineering

Course Objectives

- To learn Software development life cycle and its types.
- To know about Software Requirement Analysis & Specification**.**
- To learn about software design principles and apply them.
- To learn to apply different software testing techniques.
- To learn about Software reliability & quality assurance.

Course Outcomes

1. Understand the fundamentals of software engineering, including software development life cycle models.
2. Analyze software requirements, perform requirement analysis, and specify software requirements.
3. Apply software design principles and methodologies.
4. Evaluate coding standards, conduct testing procedures, and ensure software quality through verification and validation techniques.
5. Create software reliability, quality assurance measures, and maintenance processes.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	2	3	1	-	-	-	-	-	-	-	-	-
CO2	-	-	3	1	-	2	-	-	2	-	-	-
CO3	-	1	-	3	2	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit-I: Introduction **9 Hours**

Introduction: The product and the process, program v/s software products, Emergence of software engineering, software development life cycle models, classical waterfall, iterative waterfall, prototyping, evolution, spiral & RAP model, comparison of various life cycle models, project management process, process management process.

Unit-II: Software Requirement Analysis & Specification **9 Hours**

Software Requirement Analysis & Specification (SRAS): Need for software requirement specification, requirement process, requirement analysis, requirement specification, planning a software project; cost estimation, project scheduling, staffing & personnel planning, software configuration management, plans: quality assurance plan, risk management.

Unit-III: Software Design **9 Hours**

Software Design: Criteria for Software design, software design & design principle; module level concepts: Coupling and Cohesion, design notation & specifications, design methodology, verification, object-oriented design: Basic concepts, design methodology & Metrics, object-oriented Vs function-oriented design, detailed design.

Unit-IV: Coding and Testing **9 Hours**
Coding and Testing: Standard guideline for coding, programming practices, testing fundamentals, unit testing, verification Vs validation, black box & white box testing, functional testing, structural testing, object-oriented program testing.

Unit-V: Software reliability & quality assurance **9 Hours**
Software reliability & quality assurance: Reliability metrics, growth and modeling, software quality management system, evolution, ISO 9000. CASE: scope and benefit, support in software life cycle, CASE tools, hardware and environmental requirements, architecture of a CASE environment. Software maintenance.

Total Hours: 45

Reference Books:

1. Pankaj Jalote: An Integral Approach to Software Engineering, Narosa
2. Rogers Pressman: Software Engineering, a practitioner's approach, MGH
3. Rajib Mall: Fundamental of Software Engineering, PHI
4. Richard Farley: Software Engineering Concept, TMH
5. Hoffer "Modern System Analysis & Design" 3e, Pearson Edition

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/
3. https://www.edx.org/
4. https://www.,udemy.com

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

BCA (Cloud Computing)

Semester-V

L-3 T-0 P-0 C-3

25BCC203T: Data Warehousing and Mining

Course Objectives

- To learn basics of Data warehouse and its architecture.
- To study Data Mining fundamentals.
- To understand Data Pre-processing techniques.
- To evaluate different models used for OLAP and data pre-processing.

Course Outcomes

1. Understand the concepts of data warehousing, including its characteristics, architecture.
2. Explore data mining techniques and functionalities.
3. Analyze data preprocessing methods such as cleaning, integration etc.
4. Evaluate business analysis using data warehouse and OLAP technology.
5. Create queries to explore data warehouse

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	-	3	1	-	2	-	-	1	-	-	-	-
CO2	-	2	3	1	-	-	-	-	2	-	-	-
CO3	2	1	-	3	-	-	1	-	-	-	-	-
CO4	-	-	2	-	3	2	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit-I: Introduction to Data warehouse **9 Hours**

Data warehouse: Introduction to Data warehouse, Definition, Difference between database systems and data warehouses. Data warehouse Characteristics, Data warehouse Architecture and its Components.

Unit-II: Data Mining **9 Hours**

Data Mining: Introduction, What is Data Mining, Definition, KDD, KDD Vs Data Mining, DBMS Vs Data Mining, Classification of Data Mining System, Data Mining Functionalities, Data Mining Technique, Major Issues in Data Mining

Unit-III: Data Preprocessing **9 Hours**

Data Preprocessing: Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization and Concept Hierarchy Generation. Data Mining Primitives, Languages, and System Architectures, Concept Description: Characterization and Comparison, Analytical Characterization.

Unit-IV: Business Analysis **9 Hours**

BUSINESS ANALYSIS: Data Warehouse and OLAP Technology for Data Mining: Online Analytical Processing (OLAP), Differences between Operational Database Systems and Data

Warehouses, a multidimensional Data Model, OLAP Guidelines, Multidimensional Vs Multi relational OLAP, Data Warehouse Implementation, Data Cube Technology.

Unit-V: Cluster Analysis **9 Hours**
Cluster Analysis, Types of Data, Categorization of Major Clustering Methods, K-means, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid Based Methods, Model Based Clustering Methods, Clustering High Dimensional Data, Constraint Based Cluster Analysis, Outlier Analysis, and Data Mining Applications.

Total Hours: 45

Reference Books:

1. W. H. Inmon “Building the Data warehouse”, 3ed, Wiley India.
2. Anahory, “Data Warehousing in Real World”, Pearson Education.
3. Adriaans, “Data Mining”, Pearson Education.
4. J. Han and M. Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufmann Pub.
5. Berson “Dataware housing, Data Mining & DLAP”, @004, TMH.
6. A.K. Pujari, “Data Mining Techniques”, University Press, Hyderabad

List of e-Learning Resources:
1. https://nptel.ac.in/
2. https://www.coursera.org/
3. http://www.edx.org/
4. https://www.udemy.com

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

BCA (Cloud Computing)

Semester-V

L-0 T-0 P-4 C-2

25BCC170P: Cloud Web Services

Course Objectives

- To learn Introduction about AWS and Identity access management policy.
- Learn AWS Storage and Compute services.
- To Learn about AWS Database and Networking.
- To get knowledge about different AWS Applications Services.
- To learn Cloud Server less Designing Principles.

Course Outcomes (COs)

1. Students will be able to understand the fundamental concepts of AWS, its history, global infrastructure, and effectively manage user access using IAM.
2. Students will be able to apply AWS storage services such as S3, Glacier, and Snowball to build scalable and reliable storage solutions.
3. Students will be able to deploy and manage compute instances, configure security settings, and set up networking environments.
4. Students will be able to deploy and manage relational and NoSQL databases, understand data warehousing solutions, and implement caching mechanisms.
5. Students will be able to use AWS application services for building scalable and decoupled architectures, and integrate various services to enhance application functionality.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	-	3	1	-	2	-	-	-	-	-	-	-
CO2	1	-	3	-	-	2	-	-	2	-	-	-
CO3	2	1	-	3	-	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	1	-	-	2	1
CO5	-	1	2	-	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit-I: AWS Fundamentals and Identity Management 12 Hours

Sign up for AWS Free Tier, create and manage IAM users and policies, understand AWS infrastructure regions and availability zones, and explore AWS services and their history.

Unit-II: AWS Storage Services 12 Hours

Create an S3 bucket to upload files, implement lifecycle policies, configure Glacier for archival storage, and utilize Snowball for data transfer.

Unit-III: AWS Compute and Networking Services 12 Hours

Launch and manage EC2 Linux and Windows instances, configure security groups, EBS volumes, EFS, VPC, NAT, ACLs, firewall rules, DNS settings, register a domain, and set up routing with Route 53.

Unit-IV: AWS Database Services **12 Hours**

Deploy RDS relational databases, set up DynamoDB, explore Aurora, understand Redshift for data warehousing, and implement ElastiCache for caching.

Unit-V: AWS Application and Integration Services **12 Hours**

Utilize SQS for message queuing, configure SNS for event notifications, implement SWF for task coordination, use Elastic Transcoder for media transcoding, and set up API Gateway for building APIs.

.

Total: 60 Hours

List of Experiments

1. Sign up for AWS Free Tier.
2. Create an IAM user with appropriate permissions.
3. Explore and understand AWS Global Infrastructure regions and availability zones.
4. Set up IAM policies for user access management.
5. Create IAM groups and add users to them.
6. Learn about the history of AWS and its evolution.
7. Overview of various AWS services and their use cases.
8. Create an S3 bucket and upload files to it.
9. Implement S3 lifecycle policies to manage object storage.
10. Configure Glacier for archival storage.
11. Utilize Snowball for data transfer.
12. Understand EC2 instances and launch a Linux instance.
13. Launch a Windows EC2 instance.
14. Configure security groups for EC2 instances.
15. Create EBS volumes and snapshots.
16. Set up an Elastic File System (EFS).
17. Create and manage a Virtual Private Cloud (VPC).
18. Configure Network Address Translation (NAT) and Access Control Lists (ACLs).
19. Set up firewall rules for VPC security.
20. Configure DNS settings and register a domain name.
21. Set up routing with Route 53.
22. Deploy a relational database using RDS
23. Set up DynamoDB for NoSQL database requirements
24. Explore Aurora and its benefits
25. Understand Redshift for data warehousing
26. Implement Elasticache for caching needs
27. Utilize Simple Queue Service (SQS) for message queuing
28. Configure Simple Notification Service (SNS) for event notifications
29. Implement Simple Workflow Service (SWF) for task coordination
30. Utilize Elastic Transcoder for media transcoding
31. Configure API Gateway for building APIs

Total: 60 Hours

Reference Books:

1. AWS Documentation.

2. Amazon Web Services for Dummies, Bernald Golden, John Wiley & Sons, 2013 Reference Book.
3. Amazon Web Services Tutorial for Beginners: Bert David.
4. Brief Guide to Cloud Computing, Christopher Barnett, Constable & Robinson Limited, 2010.
5. "AWS Certified Solutions Architect - Official Study Guide: Associate Exam", Joe Baron, Hisham Baz, Tim Bixler, Biff Gaut, Kevin E. Kelly, Sean Senior, John Stamper, Sybex.
6. "Amazon Web Services in Action", Andreas Wittig and Michael Wittig, Manning Publications.
7. "AWS Certified Cloud Practitioner Study Guide: CLF-C01 Exam", Ben Piper and David Clinton, Wiley Publications.

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/
3. https://www.udemy.com/
4. https://www.edx.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

BCA(System Administration and Cyber Security)

Semester-V

L-0 T-0 P-4 C-2

25BCC191P: Programming in JAVA

Course Objectives

- To learn about the basic introduction to Java.
- To learn about classes, objects, and methods.
- To learn and practice the Collection framework.
- To know about packages.
- To know about event handling.

Course Outcomes

Students will be able to:

1. Understand the concepts of Java.
2. Apply classes, objects, and methods effectively.
3. Analyze the Java Collection framework for efficient data management.
4. Evaluate the strengths and usage of Java packages in application development.
5. Create event handling mechanisms in Java applications.

Articulation Matrix:-

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation))

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	-	3	1	-	2	-	-	-	-	-	-	-
CO2	-	1	3	2	-	-	-	-	2	-	-	-
CO3	-	2	-	3	-	-	1	-	-	-	-	-
CO4	1	2	-	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit-: Introduction to Java : **12 Hours**
History of Java, Java vs. C++, object-oriented features, JDK, JRE, JVM, comments, data types, operators, control flow (if, switch), loops (for, while, do...while), special statements, arrays.

Unit-II: CLASSES, OBJECTS AND METHODS : **12 Hours**
Introduction; Defining a Class; Adding Variables; Constructors; Methods Overloading; Static Members; Inheritance: Extending a Class; Overriding Methods; Abstract Methods and Classes; Visibility Control, Math class, Random Class.

Unit-III: Collection framework : **12 Hours**
Key Java collections and concepts: LinkedList, HashSet, TreeSet, Hashtable; Strings and

functions; ArrayList traversal (foreach, Iterator, ListIterator); Wrapper Classes; Auto Boxing/Unboxing. Interfaces: Introduction, defining, extending, implementing; accessing variables; achieving multiple inheritance.

Unit-IV: PACKAGES : 12 Hours
Introduction; System Packages; Adding a Class to a Package; Hiding Classes.
Multithreading and Exception Handling: Basic idea of multithreaded programming, the lifecycle of a thread, Basic idea of exception handling (try,catch,throw and finally).

Unit-V: Event Handling: 12 Hours
Delegation Event Model, Events, Event classes, Event listener interfaces, Abstract Window Toolkit: Window Fundamentals, Component, Container. Components – Labels, Buttons, Check Boxes, RadioButtons, Choice Menus, TextFields, Text, ScrollingList, Scrollbars, Panels, Frames. Layouts: Flow Layout, Grid Layout, Border Layout, Card Layout
Swings: Classes, Working with JFrame Windows, Working with Graphics, Working with Colour, Adding and Removing Controls, Responding to Controls, Labels, Buttons, Checkbox, Checkbox Group, Choice Control, Lists, Text Field, Text Area. Menus, Dialog Box, Handling Events.

Total: 60 Hours

List of Experiments

1. Write a hello world program and write down the steps to compile and run it in the command line.
2. Write a program for calculating factorial of given no.
3. Write a program for bitwise operators.
4. Write a program for calculating the area of a rectangle using class.
5. Write a program using a parameterized constructor.
6. Write a program for method overloading.
7. Write a program for static members.
8. Write a program for single level inheritance.
9. Write a program for multilevel inheritance.
10. Write a program for method overriding.
11. Write a program for the use of abstract methods and class.
12. Write a program for the use of string class.
13. Write a program for vector class.
14. Write a program for implementing interfaces.

15. Write a program for thread using thread class.
16. Write a program for thread using a runnable interface.
17. Write a program for sleep, yield, stop methods of thread class
18. Write a program for creating GUI using swing.
19. Write a program to handle button click events.
20. Write a program to handle creating registration forms using swing.

Total Hours 60

Reference Books:

1. Naughton & Schildt “The Complete Reference Java 2”, Tata McGraw Hill
2. D. S. Malik, "Java Programming: From Problem Analysis to Program Design",Cengage Learning
3. "Java: How to Program", Paul Deitel and Harvey Deitel, Pearson
4. "Java Swing", Marc Loy, Robert Eckstein, Dave Wood, and James Elliott, O'Reilly Media

List of e-Learning Resources:

1.https://nptel.ac.in/
2.https://www.coursera.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

BCA (Cloud Computing)

Semester-V

L-0 T-0 P-4 C-2

25BCC192P: Advanced Server-side Scripting Using Node.js

Course Objectives

- Learn basics of Node JS and to set up a development environment.
- To learn about Installing Packages and Adding dependency.
- To learn file handling and events.
- To learn database connectivity.
- To learn Node JS Template engine and RESTFul API

Course Outcomes

1. Students will be able to install, configure, and verify Node.js and its packages.
2. Students will be able to create, manage, and debug basic to intermediate Node.js applications.
3. Students will be able to implement file and data handling operations using Node.js.
4. Students will be able to develop and deploy web servers and APIs using Node.js and Express.js.
5. Students will be able to establish database connectivity and perform CRUD operations in Node.js applications.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	-	3	1	-	2	-	-	-	-	-	-	-
CO2	-	-	3	1	-	2	-	-	2	-	-	-
CO3	-1	-	2	3	-	-	1	-	-	1	-	-
CO4	2	-	1	-	3	-	-	1	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

Unit-I: Installation, Configuration, and Package Management **12 Hours**

Set up Node.js on Windows, explore the REPL terminal, and execute basic JavaScript commands. Learn to install, update, and manage packages using npm.

Unit-II: Basic to Intermediate Node.js Applications **12 Hours**

Develop and debug Node.js applications, understand the process model, and work with buffers, modules, and package dependencies. Utilize console methods, explore global objects, and create web modules.

Unit-III: File and Data Handling Operations **12 Hours**

Write, open, and delete files using Node.js file system module, and perform asynchronous file operations. Experiment with various I/O operations to understand their benefits.

Unit-IV: Web Servers and APIs Development **12 Hours**
Create and handle HTTP requests with a Node.js web server, and develop RESTful APIs using Express.js. Serve static resources, scale applications, and package them for deployment.

Unit-V: Database Connectivity and CRUD Operations **12 Hours**
Establish database connections in Node.js, retrieve data using select commands, and perform CRUD operations to update and delete records.

Total Hours: 60

List of Experiments

1. Install Node.js on Windows and verify installation.
2. Explore the Node.js REPL terminal and execute basic JavaScript commands.
3. Understand the Node.js process model and its advantages over traditional web server models.
4. Experiment with Node.js console methods and output formatting.
5. Create and execute simple functions in Node.js.
6. Work with buffers in Node.js and understand their usage.
7. Explore different module types in Node.js and their differences.
8. Create local modules and export functions/variables using module.exports.
10. Add dependencies to package.json and understand their significance.
11. Install packages globally and understand the implications.
12. Update packages using npm and manage versioning.
13. Create a basic web server using Node.js and handle HTTP requests.
14. Send various types of HTTP requests and observe server responses.
16. Write data to a file and observe the changes.
17. Perform asynchronous file writing and understand its benefits.
18. Open and delete files using Node.js file system module.
19. Experiment with other I/O operations in Node.js.
20. Utilize core Node.js debugger to debug applications.
21. Debug Node.js applications using Visual Studio Code.
23. Work with the Express.js framework to create APIs and serve dynamic content.
24. Serve static resources using Express.js middleware.
25. Establish database connectivity in Node.js using connection strings.
26. Configure and work with select commands to retrieve data from the database.
27. Update records in the database using Node.js.
28. Delete records from the database using Node.js and Express.js.
30. Explore global objects and utility modules in Node.js.
31. Create and use web modules in Node.js applications.
32. Develop a RESTful API using Express.js and understand its principles.
33. Discuss strategies for scaling Node.js applications.
34. Package a Node.js application for distribution and deployment.

Total Hours 60

Reference Books:

1. Node.js IN ACTION: by “Mike Cantelon, Marc Harter T.J., Holowaychuk, Nathan Rajlic” “Manning Publications Co.”
2. Learning Node: Moving to the Server-Side by “Shelley Powers” O’REILLY
3. "Node.js in Action" by Mike Cantelon, Marc Harter, T.J. Holowaychuk, and Nathan Rajlich
4. "Learning Node.js: A Hands-On Guide to Building Web Applications in JavaScript" by Marc Wandschneider:
5. "Node.js 8 the Right Way: Practical, Server-Side JavaScript That Scales" by Jim Wilson

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/
3. https://udemy.com/
4. http://www.edx.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

BCA (Cloud Computing)

Semester-V

L-0 T-0 P-4 C-2

25BCC193P: Web Application Development Using Python

Course Objectives

- To know about basic concepts of Python.
- To learn about Django.
- To learn and practice Django URL Pattern.
- To know about Django Form and Field validation.
- To know about Authentication.

Course Outcomes

1. Students will be able to write and understand basic Python programs.
2. Students will be able to create and manage Django models and use the Django admin tool.
3. Students will be able to implement views, URL patterns, and forms in Django.
4. Students will be able to develop, deploy, and secure Django applications.
5. Students will be able to use Django templates and combine various concepts to build full-stack projects.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	2	3	1	-	-	-	-	1	-	-	-	-
CO2	-	2	3	1	-	-	-	-	2	-	-	-
CO3	-	1	2	3	-	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	2	1	2

High-3 Medium-2 Low-1

Unit-I : Basic Python Programming **12 Hours**

Write programs demonstrating data types, if-else statements, and loops.

Unit-II : Django Models and Admin Tool **12 Hours**

Create and manage Django models, and customize the admin tool for better functionality.

Unit-III : Django Views, URL Patterns, and Forms **12 Hours**

Implement views, URL patterns, and forms, handling common patterns and advanced form features.

Unit-IV : Django Application Development and Security **12 Hours**

Develop, deploy, and secure Django applications with user authentication, caching, and CSRF protection.

Unit-V: Full-Stack Development with Django **12 Hours**

Use Django templates and combine concepts to build full-stack projects with dynamic content and user interactions.

List of Experiments

1. Write a Python program to demonstrate the use of different data types and variables.
2. Implement a program using if-else statements to determine eligibility for voting.
3. Write a program using loops to generate the Fibonacci sequence.
4. Create a Django model representing a simple entity (e.g., a student with attributes like name, roll number, etc.).
5. Use the Django admin tool to add, edit, and delete instances of the model created in the previous program.
6. Implement a view that utilizes Django's built-in features, such as handling forms or using the Django ORM for queries.
7. Design and implement URL patterns for different views in a Django project.
8. Use generic views to handle common patterns, such as displaying a list of objects.
9. Develop a small Django application using the MVT architecture, including models, views, and templates.
10. Create a Django form for user input and display appropriate messages based on form validation.
11. Implement custom field validation in a Django form.
12. Develop a user authentication system, including login and registration views.
13. Create a form that includes advanced features like file uploads or dynamic form fields.
14. Use Django sessions to store and retrieve user-specific data.
15. Deploy a simple Django project to a platform like Heroku or PythonAnywhere.
    Cache Framework:
16. Implement caching for a frequently accessed view in a Django application.
17. Implement security measures such as CSRF protection and secure password handling.
18. Customize the Django admin site, including adding custom actions or modifying the display.
19. Create and use Django templates to render HTML pages with dynamic content.
20. Combine various concepts learned to create a small full-stack project, including database interactions, user authentication, and dynamic templates.

Total 60 Hours

Reference Books:

1. Django Unleashed: by Andrew Pinkham
2. The Definitive Guide to Django: by Adrian Holovaty
3. Lightweight Django by Elman and Lavin

List of e-Learning Resources:

1.https://nptel.ac.in/
2.https://www.coursera.org/

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**

BCA (Cloud Computing)

Semester-V

L-0 T-0 P-8 C-4

25BCC230P: Minor Project

Course Objectives

- To solve industrial (or society or research) problems.
- To plan, schedule, and monitor the software project.
- To learn Development, coding, and testing of a large project cohesively.
- To learn Documentation of projects.

Course Outcomes

1. Understand fundamental concepts and principles relevant to their project domain.
2. Apply appropriate methodologies and technologies to design and develop a software solution or system addressing a real-world problem or need.
3. Analyze the effectiveness, efficiency, and quality of their software solution through testing, debugging, and optimization processes.
4. Evaluate their knowledge, skills, and experiences to innovate and propose enhancements or extensions to their project, demonstrating creativity and problem-solving abilities.
5. Develop solutions for contemporary problems using modern tools for sustainable development.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of the correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PSO1	PSO2
CO1	2	3	1	-	-	-	-	-	-	-	-	-
CO2	-	2	3	1	-	-	-	-	2	-	-	-
CO3	-	1	2	3	-	-	1	-	-	-	-	-
CO4	1	-	2	-	3	-	-	-	-	-	2	1
CO5	-	1	-	2	-	3	-	-	-	-	1	2

High-3 Medium-2 Low-1

GUIDELINES FOR BCA MINOR PROJECT

1. Every student, (in a group of maximum two) will be asked to select a particular project listed by the department, on which he/she will have to develop a working module in semesters.
2. The students are expected to work on real-life project. However, it is not mandatory for a student to work on a real-life project.
3. No student will be allowed to change the topic of the project once allotted.
4. Not more than two students are permitted to work on a project.

5. The student can formulate a project problem with the help of her/his supervisor and if approved, the students commence working on it.
6. A candidate is required to present the progress of the project work during the semester as per the schedule.

PROJECT SYNOPSIS FORMAT

The project proposal should be prepared and approved in consultation with supervisor. The project proposal should clearly state the project objectives and the environment of the proposed project to be undertaken. The project proposal should contain complete details in the following form:

1. Title of the project
2. Name of the supervisor (external supervisor(company)from / internal supervisor (teacher of the BCA)
3. Introduction and objectives of the project
4. Analysis (DFD, ER diagrams, class diagrams, time line etc. As per the project requirements).
5. A complete structure which includes:
    - Name of modules and their description
    - Database / data structures description
    - Process logic of each module (flow chart)
    - Reports generation. (repot format)
6. Tools / platform, hardware and software requirement specifications
7. Organization/ company details with profile of supervisor (if project is carried out outside the department)

PROJECT REPORT FORMULATION

Good quality white executive bond paper A4 size should be used for typing and duplication. Care should be taken to avoid smudging while duplicating the copies.

Page specification: leftmargin-3.0cms, right margin- 2.0 cm, top margin 2.54 cm, bottom margin 2.54 cm, line spacing – single, font size – 12 for normal text, 14 for headings, 16 for chapter heading, page numbers - all text pages as well as program source code listing should be numbered at the bottom of the pages. Employ MS-Word or open-source software.

The project report should contain the following:

1. Front page – black color with golden or white text.
2. Certificate from the supervisor with her/his signature and date.
3. Certificate of originality/ self-certificate.

Prepared By	**Academic Coordinator**	**HOD**	**Senior Faculty nominated by DOAA**