ACADEMIC REGULATIONS

COURSE STRUCTURE & DETAILED SYLLABUS

For

Bachelor of Computer Applications (BCA)

(Applicable for the batches admitted from 2024-25)

SREE VAHINI INSTITUTE OF SCIENCE AND TECHNOLOGY
(AUTONOMOUS)
TIRUVURU–521235, ANDHRA PRADESH, INDIA

IIBCAI SEM: DT:16-09-2025

S.No.	Category	Subject Name	Subject Code	L	T	P	Credits
1	CORE	Operating Systems	V245210031	3	0	0	3
2	CORE	Python Programming	V245210032	3	0	0	3
3	CORE	Database Management Systems	V245210033	3	0	0	3
4	CORE	Artificial Intelligence	V245210034	3	0	0	3
5	SEC	Design Thinking and Innovation	V245210071	1	1	0	2
6	CORE	Operating Systems Lab	V245210061	0	0	3	1.5
7	CORE	Python Programming Lab	V245210062	0	0	3	1.5
8	CORE	Database Management Systems Lab	V245210063	0	0	3	1.5
9	SEC	Animation Design	V245210064	0	0	3	1.5
Total				13	1	12	20

IIBCAIISEM:

S.No.	Category	Subject Name	Subject Code	L	T	P	Credits
1	CORE	Computer Networks	V245220031	3	0	0	3
2	CORE	Web Technologies	V245220032	3	0	0	3
3	CORE	Software Engineering	V245220033	3	0	0	3
4	SEC	JAVA Programming	V245220071	3	0	0	3
5	CORE	Computer Networks Lab	V245220061	0	0	3	1.5
6	SEC	Android APP Development	V245220062	0	0	3	1.5
7	SEC	JAVA Programming Lab	V245220063	0	0	3	1.5
8	SEC	Web Technologies Lab	V245220064	0	0	3	1.5
9	Internship	Summer Internship and Seminar*	V245220081	0	0	0	2
Total				12	0	12	20

*Evaluated during the END of V Sem

Controller of Examinations Principal

II Year I Sem	V245210031 –OPERATING SYSTEMS	L	T	P	C
		3	0	0	3

Course Objectives:

The main objectives of the course is to make student:

- Understand the basic concepts and principles of operating systems, including process management,memory management,file systems, and Protection.
- Make use of process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
- Illustrate different conditions for dead lock and their possible solutions.

UNIT-I:

Introduction to Operating System Concepts: Types of operating systems, operating systems concepts, Evaluation of operating systems, operating systems services, structure of OS, Introduction to System call, System call types.

UNIT-II:

Process Management: Process concept, The process, Process State Diagram, Process control block, Process Scheduling: Scheduling Queues, Schedulers, Operations on Processes, Inter process Communication, Threading Issues, Scheduling-Basic Concepts, Scheduling Criteria, Scheduling Algorithms.

UNIT-III:

Memory Management: Swapping, Contiguous Memory Allocation, Paging, structure of the Page Table,Virtual Memory Management:Virtual Memory, Demand Paging, Page-Replacement Algorithms.

UNIT-IV:

Concurrency: Process Synchronization, The Critical- Section Problem, Synchronization Hardware,Semaphores,ClassicProblems of Synchronization, Monitors, Synchronization examples Principles ofdeadlock: System Model, Deadlock Characterization, Deadlock Prevention, Detection and Avoidance, Recovery form Deadlock

UNIT-V:

File System Interface:Concep to file, Access Methods, Directory structure, File system mounting, file sharing, protection. File System Implementation: File system structure, allocation methods, Disk scheduling.

Text Book:

1. Operating System Concepts, Abraham Silberschatz,Peter Baer Galvin and Greg Gagne 9th Edition, John Wiley and Sons Inc., 2012.
2. Operating Systems–Internals and Design Principles, William Stallings, 7thEdition, Prentice Hall, 2011.

Reference Books:

1. Operating Systems: A Concept Based Approach,D.M Dhamdhere, 3rdEdition, McGraw- Hill, 2013

OnlineLearningResources:

1. https://nptel.ac.in/courses/106/106/106106144/
2. http://peterindia.net/OperatingSystems.html

II Year I Sem	V245210032 – PYTHON PROGRAMMING	L	T	P	C
		3	0	0	3

Course Objectives:

The main objectives of the course is to make student

- Understand and use scripting Language
- To expose various problem solving approaches of computer science Syllabus

UNIT–I:

Introduction: History of Python, Python Language, Features of Python, Applications of Python, Using the REPL (Shell), Running Python Scripts, Variables, Assignment, Keywords, Input

UNIT–II:

Types, Operators and Expressions:** Types: Integers, Strings, Booleans; Operators – Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators; Expressions and Order of Evaluations; Control Flow – if, if-elif-else, for, while, break, continue.

UNIT–III:

Data Structures: Lists – Operations, Slicing, Methods; Tuples, Sets, Dictionaries, Sequences, Comprehensions.

UNIT – IV:

Functions:Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Variable-length Arguments, Anonymous Functions, Fruitful Functions (Functions Returning Values), Scope of the Variables in a Function – Global and Local Variables.

UNIT – V:

Modules: Creating Modules, Import Statement, From Import Statement, Name Spacing, Python Packages, Introduction to PIP, Installing Packages via PIP, Using Python Packages.

Errors and Exceptions:Difference Between an Error and an Exception, Handling Exceptions, Try-Except Block, Raising Exceptions, User-Defined Exceptions.

Text Books:

1. Core Python Programming – W. Chun, Pearson.
2. Fundamentals of Python: First Programs – Kenneth A. Lambert, Cengage.

Reference Books:

1. Python Programming: A Modern Approach– Vamsi Kurama, Pearson.
2. Think Python – Allen Downey, Green Tea Press.
3. Introduction to Python Programming – Gowrishankar S., Veena A., CRC Press.

---Web-Resources:

https://onlinecourses.nptel.ac.in/noc20_cs83/preview

https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012664745277808640481_shared/overview

II Year I Sem	V245210033 – DATABASE MANAGEMENT SYSTEMS	L	T	P	C
		3	0	0	3

Course Objectives

The main objectives of the course are to enable students to:

Understand core database concepts, including applications, architecture, data models, and the evolution of database systems.

Design databases using ER modeling, capturing entities, relationships, and constraints, including advanced ER features.

Apply the relational model, enforce integrity constraints, and convert ER models into relational schemas.

Use SQL effectively to query, update, and manage databases, including the use of constraints, triggers, and stored procedures.

Implement normalization techniques to refine schemas, ensure data integrity, and apply various normal forms.

UNIT – I: Overview of Database System

Database System Applications, Purpose of Database Systems, View of Data, Database Languages, Relational Databases, Database Design, Data Storage and Querying, Transaction Management, Database Architecture, Data Mining and Information Retrieval, Specialty Databases, Database Users and Administrators, History of Database Systems.

[Text Book – 2]

UNIT – II: Introduction to Database Design

Database Design and ER Diagrams, Entities, Attributes and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design with the ER Model, Extended ER Features.

[Text Book – 1]

UNIT – III: Relational Model Introduction to the Relational Model, Integrity Constraints over Relations, Enforcing Integrity Constraints, Querying Relational Data, Logical Database Design – ER to Relational, Introduction to Views, Destroying/Altering Tables and Views.

[Text Book – 1]

UNIT – IV: SQL – Queries, Constraints, and Triggers

The Form of a Basic SQL Query, UNION, INTERSECT, and EXCEPT, Nested Queries, Aggregate Operators, Null Values, Complex Integrity Constraints in SQL, Triggers, Exceptions, Procedures, and Functions.

[Text Book – 1]

UNIT – V: Normal Forms

Introduction to Schema Refinement, Functional Dependencies, Reasoning about FDs, Normal Forms, Properties of Decompositions, and Normalization.

[Text Book – 1]

Text Books:

1. Database Management Systems (3/e) – Raghurama Krishnan, Johannes Gehrke, McGraw-Hill.
2. Database System Concepts (6/e) – Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.

Reference Books:

1. Database Systems(6/e) – Ramez Elmasri, Shamkant B. Navathe, Pearson.
2. Introduction to Database Systems (8/e) – C. J. Date, Pearson.
3. Database Systems (9/e) – Carlos Coronel, Steven Morris, Peter Rob, Cengage.

---Web-Resources:

1) https://nptel.ac.in/courses/106/105/106105175/
2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0127580 6667282022456_shared/overview

II Year I Sem	V245210034 – ARTIFICIAL INTELLIGENCE	L	T	P	C
		3	0	0	3

Pre-requisite

1. Knowledge in Computer Programming.
2. A course on Mathematical Foundations of Computer Science.
3. Background in Linear Algebra, Data Structures, Algorithms, and Probability.

Course Objectives

1. The student should study the concepts of Artificial Intelligence and learn methods of solving problems using AI.
2. The student should be introduced to the concepts and applications of AI, such as game playing, theorem proving, and machine learning.
3. To learn different knowledge representation techniques.

UNIT–I:Introduction AI Problems, Foundation of AI and History of AI, Intelligent Agents: Agents and Environments, The Concept of Rationality, The Nature of Environments, Structure of Agents, Problem-Solving Agents, Problem Formulation.

UNIT – II: Searching

Searching for Solutions, Uninformed Search Strategies – Breadth-First Search, Depth-First Search.
Search with Partial Information (Heuristic Search): Hill Climbing, A, AO Algorithms.
Problem Reduction, Game Playing – Adversarial Search, Games, Mini-Max Algorithm, Optimal Decisions in Multiplayer Games, Problems in Game Playing, Alpha-Beta Pruning, Evaluation Functions.

UNIT – III: Knowledge Representation

Knowledge Representation Issues, Predicate Logic – Logic Programming, Semantic Nets – Frames and Inheritance, Constraint Propagation, Representing Knowledge Using Rules, Rule-Based Deduction Systems.
Reasoning Under Uncertainty – Review of Probability, Bayes' Probabilistic Inferences, and Dempster-Shafer Theory.

UNIT – IV: Mathematical Logic

First Order Logic, Inference in First Order Logic, Propositional vs. First Order Inference, Unification and Lifting, Forward Chaining.

UNIT – V: Learning

Backward Chaining, Resolution, Learning from Observation – Inductive Learning, Decision Trees, Explanation-Based Learning, Statistical Learning Methods, Reinforcement Learning**.**

Text Books:

1. S. Russell and P. Norvig, Artificial Intelligence – A Modern Approach, Second Edition, Pearson Education.
2. Kevin Knight and Elaine Rich, Nair B., Artificial Intelligence (SIE), McGraw-Hill.

Reference Books:

1. David Poole, Alan Mackworth, Randy Goebel, Computational Intelligence: A Logical Approach, Oxford University Press.
2. G. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Fourth Edition, Pearson Education.
3. J. Nilsson, Artificial Intelligence: A New Synthesis, Elsevier Publishers.
4. Saroj Kaushik, Artificial Intelligence, Cengage Learning.

OnlineLearningResources:

1. https://ai.google/
2. https://swayam.gov.in/nd1_noc19_me71/preview

II Year I Sem	V245210061- OPERATING SYSTEMS LAB	L	T	P	C
		0	0	3	1.5

Course Objectives:

• To implement the scheduling algorithms, page replacement algorithms, file allocation methods, and understand and implement IPC mechanisms using named and unnamed pipes.

List of Experiments

1. Write C programs to simulate the following CPU scheduling algorithms:
   a) Round Robin b) SJF

2. Write C programs to simulate the following CPU scheduling algorithms:
   a) FCFS b) Priority

3. Write C programs to simulate the following file organization techniques:
   a) Single Level Directory b) Two Level c) Hierarchical

4. Write C programs to simulate the following file allocation methods:
   a) Contiguous b) Linked c) Indexed

5. Write a C program to copy the contents of one file to another using system calls.

6. Write a C program to simulate Banker's Algorithm for Deadlock Avoidance.

7. Write a C program to simulate Banker's Algorithm for Deadlock Prevention.

8. Write C programs to simulate the following page replacement algorithms:
   a) FIFO b) LRU c) LFU

II Year I Sem	V245210062 – PYTHON PROGRAMMING LAB	L	T	P	C
		0	0	3	1.5

Course Objectives

- To be able to introduce core programming basics and various operators of the Python programming language.
- To demonstrate Python data structures like Lists, Tuples, Sets, and Dictionaries.
- To understand Functions, Modules, and Regular Expressions in Python Programming.

Note: Students are encouraged to practice using Jupyter Notebook and Google Colab

List of Experiments

Exercise – 1

a) Write a program to demonstrate different number data types in Python.
b) Write a program to perform different arithmetic operations on numbers in Python.
c) Write a program to create, concatenate, and print a string and access a sub-string from a given string.

Exercise – 2

a) Write a Python script to print the current date in the following format:
WED 09 02:26:23 IST 2020"
b) Write a Python program to convert temperatures to and from Celsius and Fahrenheit.
[Formula: C/5 = (F–32)/9]
c) Write a Python script that prints prime numbers less than 20.

Exercise – 3

a) Write a Python program to find the factorial of a number using recursion.
b) Write a Python program to define a module to find Fibonacci numbers and import the module into another program.
c) Write a Python class to convert an integer to a Roman numeral.

Exercise – 4

a) Write a script named copyfile.py.This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.
b) Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.

Exercise – 5

a) Write a Python class to implement pow(x, n).
b) Write a Python class to reverse a string word by word.
c) Write a Python program to print the following triangle:

1234
234
34
4

Exercise – 6

a. Write a program to count the occurrences of each character in the string and store them in a dictionary data structure.

b.Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.

Exercise – 7

a.Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file, or a text file?

b. Write a function nearly_equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.

Exercise – 8

a.Find mean, median, and mode for the given set of numbers in a list.

b. Write a function dups to find all duplicates in the list.

c. Write a function unique to find all the unique elements of a list.

Exercise – 9

a. Write a program that defines a matrix and prints it.

b. Write a program to perform addition of two square matrices.

c. Write a program to perform multiplication of two square matrices.

II Year I Sem	V245210063 – DATABASE MANAGEMENT SYSTEMS LAB	L	T	P	C
		0	0	3	1.5

Course Objectives

- Populate and query a database using SQL DDL/DML commands.
- Declare and enforce integrity constraints on a database.
- Write queries using advanced concepts of SQL.
- Program in PL/SQL including procedures, functions, cursors, and triggers.

Experiments Covering the Topics

- DDL, DML, DCL commands
- Queries, nested queries, built-in functions
- PL/SQL programming – control structures
- Procedures, functions, cursors, triggers

Sample Experiments

1. Creation, altering and dropping of tables, and inserting rows into a table (use constraints while creating tables). Include examples using the SELECT command.

2. Queries (along with subqueries) using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT, and constraints. Example: Select the roll number and name of the student who secured fourth rank in the class.

3. Queries using aggregate functions – COUNT, SUM, AVG, MAX, and MIN; use GROUP BY, HAVING, and creation and dropping of views.

4. Queries using conversion functions (TO_CHAR, TO_NUMBER, TO_DATE), string functions (Concatenation, LPAD, RPAD, LTRIM, RTRIM, LOWER, UPPER, INITCAP, LENGTH, SUBSTR, INSTR), and date functions (SYSDATE, NEXT_DAY, ADD_MONTHS, LAST_DAY, MONTHS_BETWEEN, LEAST, GREATEST, TRUNC, ROUND, TO_CHAR, TO_DATE).

5. i. Create a simple PL/SQL program that includes declaration, executable, and exception-handling sections. (Example: Student marks can be selected from the table and printed for those who secured first class, and an exception can be raised if no records are found.)

ii. Insert data into the student table and use COMMIT, ROLLBACK, and SAVEPOINT in a PL/SQL block.

6. Develop a program that includes the features NESTED IF, CASE, and CASE expression. The program can be extended using the NULLIF and COALESCE functions.

7. Program development using WHILE LOOPS, numeric FOR LOOPS, and nested loops. Include error handling, built-in exceptions, user-defined exceptions, and RAISE-APPLICATION ERROR.

8. Program development using creation of procedures, passing parameters IN and OUT of procedures.

9. Program development using creation of stored functions, invoke functions in SQL statements, and write complex functions.

10. Develop programs using features: parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT OF clause, and CURSOR variables.

11. Develop programs using BEFORE and AFTER triggers, Row and Statement triggers, and INSTEAD OF triggers.

Text Books / Suggested Reading

1. Oracle: The Complete Reference by Oracle Press.
2. Nilesh Shah, 'Database Systems Using Oracle', PHI, 2007.
3. Rick F. Vander Lans, 'Introduction to SQL', Fourth Edition, Pearson Education, 2007.

Sample Lab Exercise 1: ER Diagram and Relational Schema Design

Objective:

To model a real-world application using ER diagrams and convert it into a relational schema.

Problem Statement:

Design an ER diagram for a Student-Course Enrollment System with the following requirements:

- A student has an ID, name, and department.
- A course has a course ID, name, and credits.
- An instructor teaches multiple courses.
- Students enroll in multiple courses.
- Keep track of which instructor teaches which course.

Instructions:

Draw an ER diagram capturing entities, attributes, and relationships. Convert the ER diagram into relational schemas with primary and foreign keys. Implement the schema in SQL.

II Year I Sem	V245210064 – ANIMATION DESIGN	L	T	P	C
		0	0	3	1.5

Course Objectives:

Animation design courses aim to equip students with the skills to create moving images, tell stories through visuals, and work with various animation techniques and software.

List of Experiments:

Experiment 1

1. Design a logo, brochure, cover letter, and visiting cards.
2. Prepare a cutout of some images using Photoshop.
3. Place a nice background for those images.

Experiment 2

4. Prepare a nice background using the gradient tool.
5. Create web banners in Adobe Flash.
6. Create a logo animation in Adobe Flash.

Experiment 3

7. Draw a cartoon animation using reference.
8. Create lip-sync to characters.
9. Use filters and special effects.

Experiment 4

10. Create a scene by using mask layers animation.
11. Create any model of a car or bike.
12. Create any model of a male or female character.

Experiment 5

13. Create a natural outdoor or indoor scene.
14. Render a frame and video of indoor and outdoor scenes.
15. Apply advanced lighting using Mental Ray render.

Experiment 6

16. Animate a day and night scene of a street with the help of lighting.
17. Create title graphics.
18. Perform video–audio synchronization.
19. Create a 30-second commercial ad.

Experiment 7

20. Create an effect of snow, rain, smoke, or water.
21. Create an effect of a bomb or explosion.

Experiment 8

22. Create a natural outdoor or indoor scene.
23. Set light for day, night, and morning.
24. Render a frame and video of indoor and outdoor scenes.

II Year I Sem	V245210071 – DESIGN THINKING AND INNOVATION	L	T	P	C
		1	1	0	2

Course Objectives

- Build mindsets & foundations essential for designers.
- Learn about the Human-Centered Design methodology and understand their real-world applications.
- Use Design Thinking for problem-solving methodology for investigating ill-defined problems.
- Undergo several design challenges and work towards the final design challenge.

Apply Design Thinking on the following Streams

- Project Stream 1: Electronics, Robotics, IOT and Sensors.
- Project Stream 2: Computer Science and IT Applications.
- Project Stream 3: Mechanical and Electrical tools.
- Project Stream 4: Eco-friendly solutions for waste management, infrastructure, safety, alternative energy sources, agriculture, environmental science and other fields of engineering.

How to Pursue the Project Work?

- The first part will be learning-based—making students embrace the methodology by exploring all the phases of design thinking through the wallet/bag challenge and podcasts.
- The second part will be more discussion-based and will focus on building some necessary skills as designers and learning about complementary material for human-centered design.
- The class will then divide into teams and they will be working with one another for about 2–3 weeks. These teams and design challenges will be the basis for the final project and presentation.
- The teams start with a Design Challenge and go through all the phases more in depth—from coming up with the right question to empathizing, ideating, prototyping, and testing.
- Outside of class, students will gather requirements, identify challenges, usability, importance, etc.
- At the end, students are required to submit the final reports, and will be evaluated by the faculty.

Tasks to be Done

Task 1: Everyone is a Designer

- Understand class objectives & harness the designer mindset.

Task 2: The Wallet/Bag Challenge and Podcast

- Gain a quick introduction to the design thinking methodology.
- Go through all stages of the methodology through a simple design challenge.
- Podcast: Observe, listen, and engage with the surrounding environment and identify a design challenge.

Task 3: Teams & Problems (Brainstorming)

- Start Design Challenge and learn about teams & problems through this.
- Foster team collaboration, find inspiration from the environment, and learn how to identify problems.

Task 4: Empathizing (Brainstorming)

- Continue Design Challenge and learn empathy.
- Learn techniques on how to empathize with users.
- Go to the field and interview people in their environments.

- Submit Activity Card.

Task 5: Ideating (Business Planning)

- Continue Design Challenge and learn how to brainstorm effectively.
- Encourage exploration and foster spaces for brainstorming.
- Submit Activity Card.

Task 6: Prototyping (Business Model Generation)

- Continue Design Challenge and learn how to create effective prototypes.
- Build tangible models and use them as communication tools.
- Start giving constructive feedback to classmates and teammates.
- Submit Activity Card.

Task 7: Testing

- Finish Design Challenge and iterate prototypes and ideas through user feedback.
- Evolve ideas and prototypes through user feedback and constructive criticism.
- Get peer feedback on individual and group performance.
- Submit Activity Card.

Task 8: Pitching

- Prepare poster to communicate your idea, how it works, why it counts, and who it benefits.
- Final Report Submission and Presentation.

Task 9: Case Study

- Make student groups. Each group will generate a business model for unsolved problems in their organization and present.

Note: The colleges may arrange for Guest Speakers from various Design Fields: Graphic Design, Industrial Design, Architecture, Product Design, Organizational Design, etc. to enrich the students with the Design Thinking Concept.

References

25. Tom Kelly, The Art of Innovation: Lessons in Creativity From IDEO, America's Leading Design Firm (Profile Books, 2002).
26. Tim Brown, Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation (HarperBusiness, 2009).
27. Jeanne Liedtka, Randy Salzman, and Daisy Azer, Design Thinking for the Greater Good: Innovation in the Social Sector (Columbia Business School Publishing, 2017).

Other Useful Design Thinking Frameworks and Methodologies

- Human-Centered Design Toolkit (IDEO): https://www.ideo.com/post/design-kit
- Design Thinking Boot Camp Bootleg (Stanford D-School): https://dschool.stanford.edu/resources/the-bootcamp-bootleg
- Collective Action Toolkit (frogdesign): https://www.frogdesign.com/wp-content/uploads/2016/03/CAT_2.0_English.pdf
- Design Thinking for Educators (IDEO): https://designthinkingforeducators.com/

II Year II Sem	V245220031 – COMPUTER NETWORKS	L	T	P	C
		3	0	0	3

Course Objectives:

The main objectives of the course is to provide an understanding of computer networking theory, including principles embodied in the protocols designed for the application layer, transport layer, network layer, and data link layer of a networking stack

UNIT–I:

Introduction: Data Communications, Networks, Network Types, Internet History, Network Models: Protocol Layering, The OSI Model, TCP/IP Protocol Suite, Introduction to Physical Layer: Transmission Impairments, Data Rate Limits, Performance.

UNIT–II:

Introduction to Data-Link- Layer: Link-Layer Addressing, Error Detection and Correction: Block Coding, Cyclic Codes, Checksum Data Link Control: Data- Link Layer Protocols, HDLC, Point-To-Point (PPP), Media Access Control (MAC): ALOHA, CSMA, CSMA/CD, CSMA/CA, Reservation, Polling, Token Passing, FDMA, TDMA, CDMA

UNIT–III:

Introduction to Network Layer: Network-Layer Services, Packet Switching, Network-Layer Performance, IPV4 Addresses, Network Layer Protocols: Internet Protocol (IP), ICMPv4, Mobile IP, Unicast Routing: Routing Algorithms, Unicast Routing Protocols, Next Generation IP: IPv6 Addressing

UNIT–IV:

Introduction to Transport Layer: Introduction, Transport-Layer Protocols, Transport-Layer Protocols: User Datagram Protocol, Transmission Control Protocol: TCP Services, TCP Features, Segment, A TCP Connection, TCP Congestion Control, Flow Control, Error Control

UNIT–V:

Application Layer: WWW, E-MAIL, Domain Name System (DNS), Quality of Service: Flow Control To Improves QoS, Integrated Services

Text Books:

1. Behrouz A .Forouzan,“DataCommunicationsandNetworking”,5th Edition, McGraw HillEducation, 2013.

Reference Books:

1. Andrew S.Tanenbaum,DavidJ.Wetherall,“Computer Networks”,5th Edition, Prentice Hall,2011.
2. LarryL. Petersonand Bruce S. Davie, “Computer Networks A System Approach”, 5th Edition,MKP, 2012.

3. James F.Kurose,KeithW.Ross,"Computer Networking,ATop-Down Approach", 5thEdition, Pearson, 2012.

Web Resources:

1. https://www.geeksforgeeks.org/computer-network-tutorials/
2. https://codescracker.com/networking/

II Year II Sem	V245220071 – JAVA PROGRAMMING	L	T	P	C
		3	0	0	3

Course Objectives

The learning objectives of this course are to:

- identify Java language components and how they work together in applications.
- learn the fundamentals of object-oriented programming in Java, including defining classes, invoking methods, and using class libraries.
- learn how to extend Java classes with inheritance and dynamic binding and how to use exception handling in Java applications.
- understand how to design applications with threads in Java.
- understand how to use Java APIs for program development.

UNIT I: Object Oriented Programming

Basic concepts, Principles, Program Structure in Java: Introduction, Writing Simple Java Programs, Elements or Tokens in Java Programs, Java Statements, Command Line Arguments, User Input to Programs, Escape Sequences, Comments, Programming Style. Data Types, Variables, and Operators: Introduction, Data Types in Java, Declaration of Variables, Type Casting, Scope of Variable, Identifier, Literal Constants, Symbolic Constants, Formatted Output with printf() Method, Static Variables and Methods, Attribute Final, Introduction to Operators, Precedence and Associativity of Operators, Assignment Operator (=), Basic Arithmetic Operators, Increment (++) and Decrement (--) Operators, Ternary Operator, Relational Operators, Boolean Logical Operators, Bitwise Logical Operators. Control Statements: Introduction, if Expression, Nested if Expressions, if–else Expressions, Ternary Operator ?:, Switch Statement, Iteration Statements, while Expression, do–while Loop, for Loop, Nested for Loop, For–Each for Loop, Break Statement, Continue Statement.

UNIT II: Classes and Objects

Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another, Access Control for Class Members, Accessing Private Members of Class, Constructor Methods for Class, Overloaded Constructor Methods, Nested Classes, Final Class and Methods, Passing Arguments by Value and by Reference, Keyword this. Methods: Introduction, Defining Methods, Overloaded Methods, Overloaded Constructor Methods, Class Objects as Parameters in Methods, Access Control, Recursive Methods, Nesting of Methods, Overriding Methods, Attributes Final and Static.

UNIT III: Arrays and Inheritance

Arrays: Introduction, Declaration and Initialization of Arrays, Storage of Array in Computer Memory, Accessing Elements of Arrays, Operations on Array Elements, Assigning Array to Another Array, Dynamic Change of Array Size, Sorting of Arrays, Search for Values in Arrays, Class Arrays, Two-dimensional Arrays, Arrays of Varying Lengths, Three-dimensional Arrays, Arrays as Vectors. Inheritance: Introduction, Process of Inheritance, Types of Inheritances, Universal Super Class - Object Class, Inhibiting Inheritance of Class Using Final, Access Control and Inheritance, Multilevel Inheritance, Application of Keyword Super, Constructor Method and Inheritance, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interfaces and Inheritance. Interfaces: Introduction, Declaration of Interface, Implementation of Interface, Multiple Interfaces, Nested Interfaces, Inheritance of Interfaces, Default Methods in Interfaces, Static Methods in Interface, Functional Interfaces, Annotations.

UNIT IV: Packages, Java Library, and Exception Handling

Packages and Java Library: Introduction, Defining Package, Importing Packages and Classes into Programs, Path and Class Path, Access Control, Packages in Java SE, Java.lang Package and its Classes, Class Object, Enumeration, Class Math, Wrapper Classes, Auto-boxing and Auto-unboxing, Java util Classes and Interfaces, Formatter Class, Random Class, Time Package, Class Instant (java.time.Instant), Formatting for Date/Time in Java, Temporal Adjusters Class. Exception Handling: Introduction, Hierarchy of Standard Exception Classes, Keywords throws and throw, try,

catch, and finally Blocks, Multiple Catch Clauses, Class Throwable, Unchecked Exceptions, Checked Exceptions. Java I/O and File: Java I/O API, standard I/O streams, types, Byte streams, Character streams, Scanner class, Files in Java (Text Book 2).

UNIT V: String Handling, Multithreading, and JDBC

String Handling in Java: Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings, Comparison, Modifying, Searching; Class String Buffer. Multithreaded Programming: Introduction, Need for Multiple Threads, Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread, Creation of New Threads, Thread States, Thread Priority, Synchronization, Deadlock and Race Situations, Inter-thread Communication - Suspending, Resuming, and Stopping of Threads. Java Database Connectivity: Introduction, JDBC Architecture, Installing MySQL and MySQL Connector/J, JDBC Environment Setup, Establishing JDBC Database Connections, ResultSet Interface.

Text Books

1. JAVA one step ahead, Anitha Seth, B.L. Juneja, Oxford.
2. Joy with JAVA, Fundamentals of Object-Oriented Programming, Debasis Samanta, Monalisa Sarma, Cambridge, 2023.
3. JAVA 9 for Programmers, Paul Deitel, Harvey Deitel, 4th Edition, Pearson.

Reference Books

1. The Complete Reference Java, 11th edition, Herbert Schildt, TMH.
2. Introduction to Java Programming, 7th Edition, Y. Daniel Liang, Pearson.

Online Resources

1. https://nptel.ac.in/courses/106/105/106105191/
2.https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012880464547618816347_shared/overview

II Year II Sem	V245220032 – WEB TECHNOLOGIES	L	T	P	C
		3	0	0	3

Course Objectives

The learning objectives of this course are to:

- identify Java language components and how they work together in applications.
- learn the fundamentals of object-oriented programming in Java, including defining classes, invoking methods, and using class libraries.
- learn how to extend Java classes with inheritance and dynamic binding and how to use exception handling in Java applications.
- understand how to design applications with threads in Java.
- understand how to use Java APIs for program development.

UNIT I: Object Oriented Programming

Basic concepts, Principles, Program Structure in Java: Introduction, Writing Simple Java Programs, Elements or Tokens in Java Programs, Java Statements, Command Line Arguments, User Input to Programs, Escape Sequences, Comments, Programming Style. Data Types, Variables, and Operators: Introduction, Data Types in Java, Declaration of Variables, Type Casting, Scope of Variable, Identifier, Literal Constants, Symbolic Constants, Formatted Output with printf() Method, Static Variables and Methods, Attribute Final, Introduction to Operators, Precedence and Associativity of Operators, Assignment Operator (=), Basic Arithmetic Operators, Increment (++) and Decrement (--) Operators, Ternary Operator, Relational Operators, Boolean Logical Operators, Bitwise Logical Operators. Control Statements: Introduction, if Expression, Nested if Expressions, if–else Expressions, Ternary Operator ?:, Switch Statement, Iteration Statements, while Expression, do–while Loop, for Loop, Nested for Loop, For–Each for Loop, Break Statement, Continue Statement.

UNIT II: Classes and Objects

Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another, Access Control for Class Members, Accessing Private Members of Class, Constructor Methods for Class, Overloaded Constructor Methods, Nested Classes, Final Class and Methods, Passing Arguments by Value and by Reference, Keyword this. Methods: Introduction, Defining Methods, Overloaded Methods, Overloaded Constructor Methods, Class Objects as Parameters in Methods, Access Control, Recursive Methods, Nesting of Methods, Overriding Methods, Attributes Final and Static.

UNIT III: Arrays and Inheritance

Arrays: Introduction, Declaration and Initialization of Arrays, Storage of Array in Computer Memory, Accessing Elements of Arrays, Operations on Array Elements, Assigning Array to Another Array, Dynamic Change of Array Size, Sorting of Arrays, Search for Values in Arrays, Class Arrays, Two-dimensional Arrays, Arrays of Varying Lengths, Three-dimensional Arrays, Arrays as Vectors. Inheritance: Introduction, Process of Inheritance, Types of Inheritances, Universal Super Class - Object Class, Inhibiting Inheritance of Class Using Final, Access Control and Inheritance, Multilevel Inheritance, Application of Keyword Super, Constructor Method and Inheritance, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interfaces and Inheritance. Interfaces: Introduction, Declaration of Interface, Implementation of Interface, Multiple Interfaces, Nested Interfaces, Inheritance of Interfaces, Default Methods in Interfaces, Static Methods in Interface, Functional Interfaces, Annotations.

UNIT IV: Packages, Java Library, and Exception Handling

Packages and Java Library: Introduction, Defining Package, Importing Packages and Classes into Programs, Path and Class Path, Access Control, Packages in Java SE, Java.lang Package and its Classes, Class Object, Enumeration, Class Math, Wrapper Classes, Auto-boxing and Auto-unboxing, Java util Classes and Interfaces, Formatter Class, Random Class, Time Package, Class Instant (java.time.Instant), Formatting for Date/Time in Java, Temporal Adjusters Class. Exception Handling: Introduction, Hierarchy of Standard Exception Classes, Keywords throws and throw, try, catch, and finally Blocks, Multiple Catch Clauses, Class Throwable, Unchecked Exceptions, Checked Exceptions. Java I/O and File: Java I/O API, standard I/O streams, types, Byte streams, Character streams, Scanner class, Files in Java (Text Book 2).

UNIT V: String Handling, Multithreading, and JDBC

String Handling in Java: Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings, Comparison, Modifying, Searching; Class String Buffer. Multithreaded Programming: Introduction, Need for Multiple Threads, Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread, Creation of New Threads, Thread States, Thread Priority, Synchronization, Deadlock and Race Situations, Inter-thread Communication - Suspending, Resuming, and Stopping of Threads. Java Database Connectivity: Introduction, JDBC Architecture, Installing MySQL and MySQL Connector/J, JDBC Environment Setup, Establishing JDBC Database Connections, ResultSet Interface.

Text Books

1. JAVA one step ahead, Anitha Seth, B.L. Juneja, Oxford.
2. Joy with JAVA, Fundamentals of Object-Oriented Programming, Debasis Samanta, Monalisa Sarma, Cambridge, 2023.
3. JAVA 9 for Programmers, Paul Deitel, Harvey Deitel, 4th Edition, Pearson.

Reference Books

1. The Complete Reference Java, 11th edition, Herbert Schildt, TMH.
2. Introduction to Java Programming, 7th Edition, Y. Daniel Liang, Pearson.

Online Resources

1. https://nptel.ac.in/courses/106/105/106105191/
2. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012880464547618816347_shared/overview.

II Year II Sem	V245220033 – SOFTWARE ENGINEERING	L	T	P	C
		3	0	0	3

Course Objectives

The objectives of this course are to introduce:

- Software lifecycle models, Software requirements and SRS document.
- Project planning, quality control and ensuring good quality software.
- Software testing strategies, use of CASE tools, implementation issues, validation & verification procedures.

UNIT I: Introduction and Software Life Cycle Models

Introduction: Evolution, Software development projects, Exploratory style of software developments, Emergence of software engineering, Notable changes in software development practices, Computer system engineering. Software Life Cycle Models: Basic concepts, Waterfall model and its extensions, Rapid application development, Agile development model, Spiral model.

UNIT II: Software Project Management and Requirements Analysis

Software Project Management: Software project management complexities, Responsibilities of a software project manager, Metrics for project size estimation, Project estimation techniques, Empirical Estimation techniques, COCOMO, risk management. Requirements Analysis and Specification: Requirements gathering and analysis, Software Requirements Specification (SRS), Formal system specification, Axiomatic specification, Algebraic specification, Executable specification and 4GL.

UNIT III: Software Design and Agility

Software Design: Overview of the design process, characterizing a good software design, Layered arrangement of modules, Cohesion and Coupling, Approaches to software design. Agility: Agility and the Cost of Change, Agile Process, Extreme Programming (XP), Other Agile Process Models, Tool Set for the Agile Process (Text Book 2). Function-Oriented Software Design: Overview of SA/SD methodology, Structured analysis, Developing the DFD model of a system, Structured design, Detailed design, and Design Review. User Interface Design: Characteristics of a good user interface, Basic concepts, Types of user interfaces, Fundamentals of component-based GUI development, and user interface design methodology.

UNIT IV: Coding, Testing, and Quality Management

Coding and Testing: Coding, Code review, Software documentation, Testing, Black-box testing, White-Box testing, Debugging, Program analysis tools, Integration testing, Testing object-oriented programs, Smoke testing, and Some general issues associated with testing. Software Reliability and Quality Management: Software reliability, Statistical testing, Software quality, Software quality management system, ISO 9000, SEI Capability maturity model, Few other important quality standards, and Six Sigma.

UNIT V: Software Maintenance and Reuse

Software Maintenance: Characteristics of software maintenance, Software reverse engineering, Software maintenance process models and Estimation of maintenance cost. Software Reuse: Reuse - definition, introduction, reason behind no reuse so far, Basic issues in any reuse program, A reuse approach, and Reuse at organization level.

Text Books

1. Fundamentals of Software Engineering, Rajib Mall, 5th Edition, PHI.
2. Software Engineering: A Practitioner's Approach, Roger S. Pressman, 9th Edition, McGraw Hill International Edition.

Reference Books

1. Software Engineering, Ian Sommerville, 10th Edition, Pearson.
2. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.

e-Resources

1) https://nptel.ac.in/courses/106/105/106105182/
2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01260589506387148827_shared/overview

II Year II Sem	V245220061 – COMPUTER NETWORKS LAB	L	T	P	C
		0	0	3	1.5

Course Objectives

Learn basic concepts of computer networking and acquire practical notions of protocols with emphasis on TCP/IP. It provides a practical approach to Ethernet/Internet networking: networks are assembled, and experiments are conducted to understand the layered architecture and how some important protocols work.

List of Experiments

1. Study of Network devices in detail and connect the computers in Local Area Network.
2. Write a Program to implement the data link layer framing methods such as:
   i) Character stuffing
   ii) Bit stuffing.
3. Write a Program to implement data link layer framing method using checksum.
4. Write a program for Hamming Code generation for error detection and correction.
5. Write a Program to implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16, and CRC CCIP.
6. Write a Program to implement Sliding Window Protocol for Go-Back-N.
7. Write a Program to implement Sliding Window Protocol for Selective Repeat.
8. Write a Program to implement Stop and Wait Protocol.
9. Write a program for congestion control using Leaky Bucket Algorithm.
10. Write a Program to implement Dijkstra's algorithm to compute the shortest path through a graph.
11. Write a Program to implement Distance Vector Routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes).

II Year II Sem	V245220062 – ANDROID APP DEVELOPMENT	L	T	P	C
		0	0	3	1.5

Course Objectives

- Understand how Android applications work, their life cycle, manifest, Intents, and using external resources.
- Design and develop useful Android applications with compelling user interfaces by using, extending, and creating your own layouts and Views and using Menus.
- Secure, tune, package, and deploy Android applications.
- Use Android's communication APIs for SMS, telephony, network management, and internet resources (HTTP).

List of Experiments

1. a) Create an android application to display “Welcome” text message.
   b) Create an android application to display “Welcome” message by using Button.
2. Create an android application to call different activities by using Implicit and Explicit Intents.
3. Create an android application to select item from given list by using Auto Complete Text View (ACTV).
4. Create an android application to display dropdown menu items and pick one item by using Spinner Component.
5. Create an android application to display internal storage data using Array Adapter.
6. Create an android application to display internal storage data in vertical format by using Custom Adapter.
7. Create an android application to display WhatsApp videos in grid view by using Custom Adapter.
8. Create an android application to display webpage by using WebView Component.
9. Create an android application to display different webpages in fragments by using Fragments Component.
10. Create an android application to store the data by using Shared Preferences.
11. Create an android application to demonstrate concept of SQLite Database storage method.
12. Create an android application to perform different types of operations (Send SMS, Make call, and Send email) by using Telephony app.
13. Write an android program to develop Media Player application.
14. Write an android program to develop Audio Recording application.
15. Write an android program to develop Video Recording application.
16. Create an android application to get latitude and longitude values by using Location Service.
17. Create an android application to get notifications on Notification Bar by using Notification Service.
18. Create an android application to display available Wi-Fi devices and paired Wi-Fi devices by using Wi-Fi Service.
19. Create an android application to get Bluetooth devices and list of devices using Bluetooth and Vibrator Service.
20. Create an android application to display current location on Google Maps by using Google Maps Service.

Reference Books

1. Android Application Development (with KitKat Support), Black Book, Pradeep Kothari.
2. Beginning Android 4 Application Development, Wei-Meng Lee.
3. Android Application Development for Dummies, Michael Burton.

II Year II Sem	V245220063 – JAVA PROGRAMMING LAB	L	T	P	C
		0	0	3	1.5

Course Objectives

The aim of this course is to:

- Practice object-oriented programming in the Java programming language.
- Implement Classes, Objects, Methods, Inheritance, Exception, Runtime Polymorphism, and User-defined Exception handling mechanism.
- Illustrate inheritance, Exception handling mechanism, and JDBC connectivity.
- Construct Threads, Event Handling, and implement packages.

Sample Experiments

Exercise – 1:
a) Write a JAVA program to display default value of all primitive data types of JAVA.
b) Write a JAVA program that displays the roots of a quadratic equation $ax^2 + bx = 0$.
Calculate the discriminant D and, based on the value of D, describe the nature of the roots.

Exercise – 2:
a) Write a JAVA program to search for an element in a given list of elements using binary search mechanism.
b) Write a JAVA program to sort elements in a given list using bubble sort.
c) Write a JAVA program using StringBuffer to delete and remove characters.

Exercise – 3:
a) Write a JAVA program to implement class mechanism. Create a class, methods, and invoke them inside the main method.
b) Write a JAVA program that implements method overloading.
c) Write a JAVA program to implement a constructor.
d) Write a JAVA program to implement constructor overloading.

Exercise – 4:
a) Write a JAVA program to implement Single Inheritance.
b) Write a JAVA program to implement Multi-level Inheritance.
c) Write a JAVA program for an abstract class to find areas of different shapes.

Exercise – 5:
a) Write a JAVA program giving an example for the "super" keyword.
b) Write a JAVA program to implement Interface.
c) Write a JAVA program that implements Runtime Polymorphism.

Exercise – 6:
a) Write a JAVA program that describes the exception handling mechanism.
b) Write a JAVA program illustrating Multiple catch clauses.
c) Write a JAVA program for creation of Java Built-in Exceptions.
d) Write a JAVA program for creation of User Defined Exceptions.

Exercise – 7:
a) Write a JAVA program that creates threads by extending Thread class.
First thread displays "Good Morning" every 1 sec, the second displays "Hello" every 2 seconds,
and the third displays "Welcome" every 3 seconds. (Repeat the same by implementing Runnable.)
b) Write a program illustrating isAlive() and join().
c) Write a JAVA program that imports and uses user-defined packages.

Exercise – 8:
a) Write a JAVA program that connects to a database using JDBC.
b) Write a JAVA program to connect to a database using JDBC and insert values into it and delete values from it.

II Year II Sem	V245220064 – WEB TECHNOLOGIES LAB	L	T	P	C
		0	0	3	1.5

Course Objectives

• To implement the web pages using HTML and apply styles.
• Able to develop a dynamic webpage by the use of JavaScript.
• Design to create structure of webpage, to store the data in web document, and transport information through web.
• Able to write a well formed/valid XML document.

Experiment 1

Develop static pages (using HTML and CSS) of an online book store. The pages should resemble: www.flipkart.com. The website should consist of the following pages:
a) Homepage
b) Registration and user Login
c) User Profile Page
d) Books catalog
e) Shopping Cart
f) Payment By credit card
g) Order Conformation

Experiment 2

Create and save an XML document on the server, which contains 10 users' information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.

Experiment 3

Write a PHP script that reads data from one file and writes it into another file.

Experiment 4

Write a PHP script to print prime numbers between 1-50.

Experiment 5

Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.

Experiment 6

Write a PHP script to:
a. Find the length of a string.
b. Count number of words in a string.
c. Reverse a string.
d. Search for a specific string.

Experiment 7

Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.

Experiment 8

Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.

Experiment 9

Install a database (MySQL or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number (these should hold the data from the registration form). Practice 'JDBC' connectivity. Write a Java program/servlet/JSP to connect to that database and extract data from the tables and display them. Experiment with various SQL queries. Insert the details of the users who register with the website, whenever a new user clicks the submit button in the registration page.

Experiment 10

Write a JSP which does the following job: Insert the details of the 3 or 4 users who register with the website (week

9) by using registration form. Authenticate the user when he submits the login form using the user name and password from the database.

***