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

ACADEMIC REGULATIONS

COURSE STRUCTURE & DETAILED SYLLABUS

For

Bachelor of Computer Applications (BCA)

(Applicable for the batches admitted from 2024-25)

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

COURSE STRUCTURE

II Year – I SEMESTER						
S. No	**Category**	**Courses**	**L**	**T**	**P**	**C**
1	CORE	Operating Systems	3	0	0	3
2	CORE	Python Programming	3	0	0	3
3	CORE	Database Management Systems	3	0	0	3
4	CORE	Artificial Intelligence	3	0	0	3
5	CORE	Operating Systems Lab	0	0	3	1.5
6	CORE	Python Programming Lab	0	0	3	1.5
7	CORE	Database Management Systems Lab	0	0	3	1.5
8	SEC	Animation Design	0	0	3	1.5
9	SEC	Design Thinking and Innovation	1	1	0	2
Total Credits						**20**

II Year – II SEMESTER						
S. No	**Category**	**Courses**	**L**	**T**	**P**	**C**
1	CORE	Computer Networks	3	0	0	3
2	SEC	JAVA Programming	3	0	0	3
3	CORE	Web Technologies	3	0	0	3
4	CORE	Software Engineering	3	0	0	3
5	CORE	Computer Networks Lab	0	0	3	1.5
6	SEC	Android APP Development	0	0	3	1.5
7	SEC	JAVA Programming Lab	0	0	3	1.5
8	SEC	Web Technologies Lab	0	0	3	1.5
9	SI	Summer Internship and Seminar*	-	-	-	2
Total Credits						**20**

* Evaluated during the END of V Sem

II Year – I Sem	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 deadlock 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, Classic Problems of Synchronization, Monitors, Synchronization examples **Principles of deadlock**: System Model, Deadlock Characterization, Deadlock Prevention, Detection and Avoidance, Recovery form Deadlock

UNIT-V:
File System Interface: Concept of a 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, 7th Edition, Prentice Hall, 2011.

Reference Books:

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

Online Learning Resources:

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

II Year – I Sem	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 problems 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-Output, Indentation.

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 (Function 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 Error and Exceptions: Difference between an error and Exception, Handling Exception, 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	DATABASE MANAGEMENT SYSTEMS	L	T	P	C
		3	0	0	3

Course Objectives:
The main objectives of the course is to make student:

- 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 Informational 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, 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, Functions **[Text Book -1]**

UNIT –V:
Normal Forms: Introduction to Schema Refinement, Functional Dependencies, Reasoning about FDs, Normal Forms, Properties of Decompositions, Normalization. **[Text Book -1]**

Text Books:

1. Data base Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke, Mc Graw-Hill
2. Data base System Concepts, 6/e, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Mc Graw-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_01275806667282022456_shared/overview

II Year I Sem	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 be made to study the concepts of Artificial Intelligence, learn the methods of solving problems using Artificial Intelligence.
2. The student should be made to introduce the concepts of applications of AI, namely 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, uniformed search strategies – Breadth first search, depth first Search. Search with partial information (Heuristic search) Hill climbing, A* ,AO* Algorithms, Problem reduction, Game Playing-Adversial search, Games, mini-max algorithm, optimal decisions in multiplayer games, Problem 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, rules based deduction systems. Reasoning under uncertainty, review of probability, Bayes' probabilistic interferences and dempstershafer theory.

UNIT – IV:

Mathematical Logic: First order logic. Inference in first order logic, propositional vs. first order inference, unification & lifts forward chaining,

UNIT – V:

Backward chaining, Resolution, Learning from observation Inductive learning, Decision trees, Explanation based learning, Statistical Learning methods, Reinforcement Learning.

Textbooks:

1. S. Russel and P. Norvig, "Artificial Intelligence – A Modern Approach", SecondEdition, Pearson Education.
2. Kevin Night and Elaine Rich, Nair B., "Artificial Intelligence (SIE)", Mc Graw 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. Artificial Intelligence, Saroj Kaushik, CENGAGE Learning.

Online Learning Resources:

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

II Year I Sem	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, understand and implement IPC mechanism 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 Bankers Algorithm for Dead Lock Avoidance
7. Write a C program to simulate Bankers Algorithm for Dead Lock Prevention
8. Write C programs to simulate the following page replacement algorithms:
   a) FIFO b) LRU c) LFU

II Year I Sem	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 Python programming language.
- To demonstrate about Python data structures like Lists, Tuples, Sets and dictionaries
- To understand about 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 accessing 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, 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 factorial of a number using Recursion.
b. Write a python program to define a module to find Fibonacci Numbers and import the module to 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

```
1 2 3 4
  2 3 4
    3 4
```

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?
a. 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, 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
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	DATABASE MANAGEMENT SYSTEMS LAB	L	T	P	C
		0	0	3	1.5

Course Objectives:
This Course will enable students to

- Populate and query a database using SQL DDL/DML Commands
- Declare and enforce integrity constraints on a database
- Writing Queries using advanced concepts of SQL
- Programming 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), examples using SELECT command.
2. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, 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), GROUP BY, HAVING and Creation and dropping of Views.
4. Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), 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 which includes declaration section, executable section and exception –Handling section (Ex. 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 were found)
   ii. Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in 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, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, 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 Exercise1: 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	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, visiting cards.
2. Prepare a cutout of some images using Photoshop.
3. Place nice background for those images.

Experiment 2:
1. Prepare nice background using gradient tool.
2. Creating Web Banners in Adobe Flash
3. Creating a Logo Animation in Adobe Flash

Experiment 3:
1. Draw Cartoon Animation using reference.
2. Create Lip Sink to Characters
3. Using filters & Special effects

Experiment 4:
1. Create a scene by using Mask layers animation
2. Create any Model of Cars or Bike,
3. Create any model of the male or female character.

Experiment 5:
1. Create a natural outdoor or indoor scene.
2. Render a frame and video of indoor and outdoor scenes.
3. Advance lighting using mental ray render.

Experiment 6:
1. Animate day and night scene of a street with the help of lighting.
2. Title Graphics
3. Video – Audio synchronization
4. 30 Second Commercial AD

Experiment 7:
1. Creating an effect of snow or rain or smoke or water.
2. Creating an effect of bomb/explosion.

Experiment 8:
1. Create a natural outdoor or indoor scene.
2. Set light for Day, Night and Morning
3. Render a frame and video of indoor and outdoor scenes.

II Year I Sem	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 to

- Project Stream 1: Electronics, Robotics, IOT and Sensors
- Project Stream 2: Computer Science and IT Applications
- Project Stream 3: Mechanical and Electrical tools
- Project Stream4: 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-masking students to 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 final presentation to be presented.
- The teams start with **Design Challenge** and go through all the phases more in depth from coming up with the right question to empathizing to ideating to prototyping and to testing.
- Outside of class, students will also be gathering the requirements, identifying the 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 (Brain Storming)

- 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 (Brain Storming)

- 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 students 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 Design Thinking Concept.

References:

1. Tom Kelly, *The Art of Innovation: Lessons in Creativity From IDEO, America's Leading Design Firm* (Profile Books, 2002)
2. Tim Brown, Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation (HarperBusiness, 2009)
3. 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/wpcontent/
  - uploads/2016/03/CAT_2.0_English.pdf
- Design Thinking for Educators (IDEO); https://designthinkingforeducators.com/

II Year II Sem	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, “Data Communications and Networking”, 5th Edition, McGraw HillEducation, 2013.

Reference Books:
1. Andrew S. Tanenbaum, David J. Wetherall, “Computer Networks”, 5th Edition, Prentice Hall,2011.
2. Larry L. Peterson and Bruce S. Davie, “Computer Networks A System Approach”, 5th Edition,MKP, 2012.

3. James F. Kurose , Keith W. Ross, " Computer Networking, A Top-Down Approach", 5thEdition, Pearson, 2012.

Web Resources:

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

II Year II Sem	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, 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, Data Types, 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: 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 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, 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 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, DebasisSamanta, MonalisaSarma, Cambridge, 2023.
3. JAVA 9 for Programmers, Paul Deitel, Harvey Deitel, 4th Edition, Pearson.

References 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_01288046454761881634 7_shared/overview

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

Course Objectives:

- To Learn PHP language for server side scripting
- To introduce XML and processing of XML Data with Java
- To introduce Server side programming with Java Servlets and JSP
- To introduce Client side scripting with JavaScript.

Unit I:

Web Basics- Introduction, Concept of Internet- History of Internet, Protocols of Internet, World Wide Web, URL, Web Server, Web Browser. **HTML- Introduction**, History of HTML, Structure of HTML Document: Text Basics, Structure of HTML Document: Images and Multimedia, Links and webs, Document Layout, Creating Forms, Frames and Tables, Cascading style sheets.

Unit II:

XML Introduction- Introduction of XMLXML: Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemes, Document Object Model, XHTML Parsing XML Data – DOM and SAX Parsers in java.

Unit III:

Introduction to Servlets: Common Gateway Interface (CGI), Life cycle of a Servlet, deploying a Servlet, The Servlet API, Reading Servlet parameters, Reading Initialization parameters, Handling Http Request & Responses, Using Cookies and Sessions, connecting to a database using JDBC.

Unit IV:

Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Using Cookies and session for session tracking, connecting to database in JSP.

Client-side Scripting: Introduction to JavaScript, JavaScript language – declaring variables, scope of variables, functions. event handlers (onClick, on Submit etc.), Document Object Model, Form validation.

Unit V:

Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads. Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies File Handling in PHP: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

Text Books:

1. Web Technologies, Uttam K Roy, Oxford University Press.
2. The Complete Reference PHP — Steven Holzner, Tata McGraw-Hill.

Reference Books:

1. Web Programming, building internet applications, Chris Bates 2″ edition, Wiley Dreamtech.
2. Java Server Pages —Hans Bergsten, SPD O'Reilly.
3. Java Script, D.Flanagan
4. Beginning Web Programming-Jon Duckett WROX.

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

Course Objectives:
The objectives of this course are to introduce

- Software life cycle 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: 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: 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: 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 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: 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, Mc-Graw 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
3) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01338269041100390473 5_shared/overview

II Year II Sem	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 the emphasis on TCP/IP. It provides a practical approach to Ethernet/Internet networking: networks are assembled, and experiments are made to understand the layered architecture and how do 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 farming methods such as i) Character stuffing ii) bit stuffing.
3. Write a Program to implement data link layer farming method 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 Goback 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	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 Web view 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, Making call and sending 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 value by using Location Service.
17. Create an android application to get the 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 the 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	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, User defined Exception handling mechanism
- Illustrate inheritance, Exception handling mechanism, JDBC connectivity
- Construct Threads, Event Handling, implement packages

Sample Experiments:

Exercise – 1:

a) Write a JAVA program to display default value of all primitive data type of JAVA
b) Write a java program that display the roots of a quadratic equation $ax^2+bx=0$. Calculate the discriminate D and basing on value of D, describe the nature of root.

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 for an element in a given list of elements using bubble sort
c) Write a JAVA program using String Buffer to delete, remove character.

Exercise - 3

a) Write a JAVA program to implement class mechanism. Create a class, methods and invoke them inside main method.
b) Write a JAVA program implements method overloading.
c) Write a JAVA program to implement 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 abstract class to find areas of different shapes

Exercise - 5

a) Write a JAVA program give example for "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 exception handling mechanism
b) Write a JAVA program Illustrating Multiple catch clauses
- Write a JAVA program for creation of Java Built-in Exceptions
- Write a JAVA program for creation of User Defined Exception

Exercise - 7

a) Write a JAVA program that creates threads by extending Thread class. First thread display "Good Morning "every 1 sec, the second thread displays "Hello "every 2 seconds and the third display "Welcome" every 3 seconds, (Repeat the same by implementing Runnable)
b) Write a program illustrating **is Alive** and **join ()**
c) Write a JAVA program that import and use the 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	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 Java Script.
- Design to create structure of web page, 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 the following pages.

a) Home page
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 write 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 no 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 web site, 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 web site (week9) by using registration form. Authenticate the user when he submits the login form using the user name and password from the database.

***