



School of Computer Science & IT Devi Ahilya Vishwavidyalaya

SYLLABUS

M.Sc (Computer Science) 2 years

Program Educational Objectives (PEOs)

- PEO 1:** Equip students with a strong foundation of computer science, emphasizing the application of theoretical and practical knowledge to design, develop, and implement efficient software solutions.
- PEO 2:** Imbibe effective research, academic, team building and communication skills, to conduct impactful work in computer science domains.
- PEO 3:** Enable continuous professional growth in emerging trends, tools, and technologies, to build successful careers in industry, academia, or entrepreneurship.

Program Specific Outcomes (PSOs)

- PSO 1:** Apply advanced knowledge in core areas of computer science, such as algorithms, database, software engineering, compiler design, networking and data science to analyse, design, and develop robust computing solutions.
- PSO 2:** Empower students to innovate in technology-driven domains, contributing to the creation of solutions to address complex computing challenges with professional ethics and social responsibilities.

IV - SEMESTER

CS-5216: DESIGN AND ANALYSIS OF ALGORITHMS

Course Outcomes (COs):

CO1: Understand complexity representation in terms of asymptotic notations.

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

CO3: Compare and analyse different data structures and sorting algorithms.

CO4: Understand and analyse major graph algorithms

CO5: Understand and apply different algorithm design paradigms such as greedy, dynamic programming, backtracking, branch and bound for effective problem-solving.

Course Contents

UNIT-I

No. of Hours: 8

Introduction: Algorithms, analysing algorithms, Complexity of algorithms, Growth of functions, Asymptotic Notations, Recurrence Relations and their Solution Methods.

UNIT-II

No. of Hours: 8

Sorting and Order Statistics - Divide-Conquer approach with Quick sort, Merge sort, Comparison of sorting algorithms, sorting in linear time (Counting, Radix, Bucket sort), Heap Sort.

Advanced Data Structures: Red-Black Tree-Properties, Insertion, B-Trees-Creation, Insertion, and Deletion. Introduction to Binomial Heaps - Merge, Union Operation, Fibonacci Heaps - Insertion, Finding Minimum Key, Union.

UNIT-III

No. of Hours: 8

Graph: Graph Traversal-Breadth First Search, Depth First Search

Greedy Methods: Fractional Knapsack, Activity Selection Problem. Minimum Spanning trees – Prim's and Kruskal's algorithms. Single source shortest paths - Dijkstra's algorithms.

UNIT-IV

No. of Hours: 8

Dynamic Programming: 0/1 Knapsack, Longest Common Subsequence (LCS), Matrix Chain Multiplication. Edit distance, All pair shortest paths – Warshal's and Floyd's algorithms, Optimal binary search tree, Bellman-Ford algorithms.

UNIT-V

No. of Hours: 8

Backtracking: Graph Coloring, n-Queen Problem, Sum of Subset, Branch and Bound: Travelling Salesman Problem, Introduction to P, NP, NP-complete, NP-Hard.

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest (2022), Introduction to Algorithms, Fourth edition, Prentice Hall of India.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran (2008), "Fundamentals of Computer Algorithms", Orient Longman Pvt. Ltd.

Reference Books:

1. Gilles Brassard Paul Bratley (1996), "Fundamentals of Algorithms", Prentice Hall.
2. Anany Levitin (2009), "An Introduction to Design and Analysis of Algorithms", Pearson.

Online Resources:

1. NPTEL Course Video Lectures on "Design and Analysis of Algorithms" - By Prof. Abhiram G Ranade, Prof. Ajit A Diwan, Prof. Sundar Viswanathan IIT Bombay
2. NPTEL Course Video Lectures on "Design and Analysis of Algorithms" - Prof. Madhavan Mukund, Chennai Mathematical Institute.

CS-5216: DESIGN AND ANALYSIS OF ALGORITHMS - PRACTICAL

Course Outcomes (COs):

CO1: Understand complexity representation in terms of asymptotic notations.

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

CO3: Compare and analyse different data structures and sorting algorithms.

CO4: Understand and analyse major graph algorithms.

CO5: Understand and apply different algorithm design paradigms such as greedy, dynamic programming, backtracking, branch and bound for effective problem-solving.

Course Contents

UNIT-I

No. of Hours: 8

Implement the following sorting algorithm for a given set of elements and determine the time required to sort the elements. The elements can be read from a file or can be generated using the random number generator.

- Quick sort
- Merge sort
- Counting, Radix, Bucket sort
- Heap Sort

UNIT-II

No. of Hours: 8

Write programs to implement the following data structures:

- Red-Black Tree
- B-Trees

UNIT-III

No. of Hours: 8

Write programs to print all the nodes reachable from a given starting node in a graph using the following traversal methods:

- Breadth First Search (BFS)
- Depth First Search (DFS)

Write programs to find the optimal solution for the following problems using the Greedy Method:

- Fractional Knapsack
- Activity Selection Problem.
- Minimum Spanning trees – Prim's and Kruskal's algorithms.
- Single source shortest paths - Dijkstra's algorithms

UNIT-IV

No. of Hours: 8

Write programs to find the optimal solution for the following problems using the Dynamic Programming approach:

- 0/1 Knapsack
- Longest Common Subsequence (LCS)
- Matrix Chain Multiplication
- Edit distance
- All pair shortest paths – Warshal’s and Floyd’s algorithms
- Optimal binary search tree
- Bellman-Ford algorithms

UNIT-V

No. of Hours: 8

Write programs to find the optimal solution for the following problems using the Backtracking approach:

- Graph Coloring
- n-Queen Problem
- Sum of Subset

Write a program to find the optimal solution for the following problems using the Branch and Bound method:

- Travelling Salesman Problem

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest (2022), Introduction to Algorithms, Fourth edition, Prentice Hall of India
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran (2008), “Fundamentals of Computer Algorithms”, Orient Longman Pvt. Ltd.

Reference Books:

1. Gilles Brassard Paul Bratley (1996),” Fundamentals of Algorithms”, Prentice Hall.
2. Anany Levitin (2009), “An Introduction to Design and Analysis of Algorithms”, Pearson.

Online Resources:

1. NPTEL Course Video Lectures on "Design and Analysis of Algorithms" - By Prof. Abhiram G Ranade, Prof. Ajit A Diwan, Prof. Sundar Viswanathan IIT Bombay
2. NPTEL Course Video Lectures on "Design and Analysis of Algorithms" - Prof. Madhavan Mukund, Chennai Mathematical Institute.

CS-5512: COMPILER DESIGN

Course Outcomes (COs):

CO1: Understand the design and functioning of Lexical Analyzers.

CO2: Develop syntax analyzers using top-down parsing and tools like YACC and LEX.

CO3: Implement efficient bottom-up parsers, including SLR, Canonical LR, and LALR.

CO4: Conduct semantic analysis and create intermediate representations.

CO5: Apply optimization techniques and manage effective code generation.

Course Contents

UNIT-I

No. of Hours: 8

Compiler Structure: Compilers and Translators, Various Phases of Compiler, Lexical Analysis: The role of Lexical Analyzer, A simple approach to the design of Lexical Analyzer, Implementation of Lexical Analyzer.

UNIT-II

No. of Hours: 8

LMD (Leftmost Derivation), RMD (Rightmost Derivation), Derivation and Parse Tree, Ambiguity, Capabilities of CFG. Basic Parsing Techniques: Top-Down parsers with backtracking, Non-Recursive Predictive Descent Parser, YACC (Yet Another Compiler Compiler), LEX utility tool.

UNIT-III

No. of Hours: 8

Bottom-up Parsers, Shift-Reduce Parsing, Operator Precedence Parsers, LR Parsers: SLR (Simple LR), Canonical LR (Left-to-right), LALR (Look-Ahead LR).

UNIT-IV

No. of Hours: 8

Semantic Analysis, Memory overflow underflow, Type matching, mismatching, global local variable. Intermediate Code Generation: Different Intermediate forms: three address code, Quadruples & Triples. Syntax Directed translation mechanism. Control flow, syntax tree, postfix translation.

UNIT-V

No. of Hours: 8

Optimization and Code Generation: Local optimization, Loop optimization, Basic blocks and flow graphs, DAG, Data flow analyzer, Symbol Table management, Error handler.

Text Books:

1. Principles of Compiler Design - Alfred V. Aho, Jeffrey D. Ullman, Narosa Publishing House.

Reference Books:

1. Compiler Construction: Principles and Practice - Kenneth C. Loudon, 1st Edition, Cengage Learning.
2. Compiler Design in C - A. C. Holub, Prentice-Hall Inc., 1993.
3. Compiler Design - Raghavan, TMH Publications.

Online Resources:

1. https://onlinecourses.nptel.ac.in/noc25_cs13/preview
2. <https://nptel.ac.in/courses/106106237>

CS-5512: COMPILER DESIGN - PRACTICAL

Course Outcomes (COs):

CO1: Implement lexical analysis using NFAs, DFAs, and LEX tools for grammar-based input.

CO2: Apply parsing techniques to compute FIRST/FOLLOW sets and design predictive parsers.

CO3: Construct bottom-up parsers and generate parse trees using JFLAP and YACC.

CO4: Develop intermediate code generators for expressions using syntax-directed translation.

CO5: Perform code optimization and manage symbol tables effectively.

Course Contents

UNIT-I

No. of Hours: 4

Lexical Analysis and Compiler Structure:

1. File Analysis:

- Develop a program in C/Java/Python to scan and count the number of characters, words, and lines in a file. (Practical foundation for lexical analysis.)

2. Design and Simulation of NFAs:

- Create a program in C/Java/Python to design and simulate an NFA capable of recognizing identifiers, keywords, constants, and comments. Accept user-defined grammar rules to handle any input language.

3. Design and Simulation of DFAs:

- Implement a program in C/Java/Python to simulate a DFA that recognizes identifiers, constants, and operators. Allow dynamic input for rules.

4. Study of LEX Tool:

- Use LEX to create a lexical analyzer for any user-defined grammar or mini-language.

UNIT-II

No. of Hours: 6

Parsing Techniques:

1. FIRST and FOLLOW Computation:

- Write a program in C/Java/Python to calculate the FIRST and FOLLOW sets for any grammar entered by the user.

2. Grammar Transformation:

- Develop a program in C/Java/Python to:
 - Eliminate left recursion from any user-defined grammar.
 - Perform left factoring on the grammar for predictive parsing.

3. Predictive Parsing:

- Write a program in C/Java/Python to implement a non-recursive predictive descent parser for any LL (1) grammar provided.

4. JFLAP-Based Experiment:

- Using JFLAP, create an LL (1) parse table for any CFG provided by the user.

UNIT-III**No. of Hours: 8**

Bottom-Up Parsing

1. SLR Parsing Using JFLAP:

- Utilize JFLAP to construct an SLR (1) parse table for a user-defined CFG.

2. Operator Precedence Parsing:

- Write a program in C/Java/Python to calculate Leading and Trailing sets for any grammar, and use these sets for operator precedence parsing.

3. Study of YACC Tool:

- Use YACC to build a parser for user-defined grammars and generate corresponding parse trees.

UNIT-IV**No. of Hours: 4**

Intermediate Code Generation

1. Intermediate Code Generation:

- Write a program in C/Java/Python to generate three-address code for arithmetic and logical expressions. The program should accept any valid expression as input.

2. Syntax-Directed Translation Mechanism:

- Implement a program in C/Java/Python to convert infix expressions to postfix or syntax trees using syntax-directed translation techniques.

UNIT-V**No. of Hours: 4**

Code Optimization and Symbol Tables

1. Code Optimization:

- Write a program in C/Java/Python to perform basic code optimization techniques such as constant folding and dead code elimination.

2. Symbol Table Management:

- Create a program in C/Java/Python to simulate a symbol table with operations like insertion, deletion, and lookup for variables.

Text Books:

1. Compiler Design in C - Allen I. Holub, Prentice-Hall.

Reference Books:

1. Compiler Construction: Principles and Practice - Kenneth C. Louden, 1st Edition, Cengage Learning.
2. Compiler Design in C - A. C. Holub, Prentice-Hall Inc., 1993.
3. Compiler Design - Raghavan, TMH Publications.

Online Resources:

1. https://onlinecourses.nptel.ac.in/noc26_cs56/preview

CS-5617: INTERNET AND WEB TECHNOLOGY

Course Outcomes (COs):

CO1: Use syntax and semantics of java programming language.

CO2: Develop reusable programs using the concepts of inheritance, polymorphism, interfaces and packages.

CO3: Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes.

CO4: Developing application using advance java concepts.

CO5: Understanding the concepts of MVC architecture.

Course Contents

UNIT-I

No. of Hours: 8

Features of Java, Object-oriented programming overview, Introduction of Java Technologies, How to write simple Java programs, Data Types, Variables, Memory concepts, control statements, looping, Method CallStack and Activation Record, Argument Promotion and Casting, Scope of declaration and Method Overloading, StringHandling: The String constructors, String operators, Character Extraction, String comparison, String Buffer. Arrays: Declaring and Creating Arrays, Enhanced for Statement, Passing Arrays to Method, Multidimensional Arrays, Variable-Length Argument lists, Using Command-line Arguments

UNIT-II

No. of Hours: 8

Inheritance: Extending classes & related things, Packages and Interfaces: Defining a Package, Understanding CLASSPATH, Access Protection, Importing packages, creating own packages Exception Handling: Introduction, overview of doing it and keywords used, when to use it, Multithreading: What are threads, The java Thread model, Thread priorities, Thread life cycle, Thread Synchronization, Applets: Applet basics, Applet Architecture, Applet life cycle methods, Database connectivity: JDBC, The design of JDBC, Typical uses of JDBC

UNIT-III

No. of Hours: 8

Introduction to HTTP, webServer and application Servers, Installation of Application servers, Config files, Web.xml. JavaServlet, Servlet Development Process, Deployment Descriptors, Generic Servlet, Lifecycle of Servlet, Servlet Packages, Classes, Interfaces, and Methods, Handling Forms with Servlet, Various methods of Session Handling, various elements of deployment descriptors.

UNIT-IV

No. of Hours: 8

JSP Basics: JSP lifecycle, Directives, scripting elements, standard actions, implicit objects. Connection of JSP and Servlet with different database viz. Oracle, MS-SQL Server, MySQL, java.sql Package, Querying database, adding records, deleting records, modifying records, types of Statement,

Separating Business Logic and Presentation Logic, Building and using JavaBean, Session handling in JSP, Types of errors and exceptions handling.

UNIT-V

No. of Hours: 8

MVC Architecture Introduction to Remote Method Invocation, Introduction to Enterprise Java Bean, Types of EJB, Creating and working with Session Bean

Text Books:

1. Java2:The Complete Reference by Herbert Schildt, Tata McGraw-Hill, 8th Edition, 2011.
2. K. Mukhar, "Beginning Java EE5: From Novice to Professional", Wrox Press.

Reference Books:

1. The Java Programming Language, Ken Arnold, James Gosling, David Holmes, 3rd Edition, Person Education, 2000
2. Head First Java, Kathy, Sierra, Bert Bates, O'Reilly Publication, 2nd Edition, 2005

Online Resources:

1. NPTEL Course Video Lectures on "Programming in java", By Prof. Debasis Samant, IIT Kharagpur
2. Swayam Course Video Lectures on "Web Technology", By Prof. Dr. Ashutosh Kumar Bhatt

CS-5617: INTERNET AND WEB TECHNOLOGY-PRACTICAL

Course Outcomes (COs):

CO1: Able to write real world problem using core java concepts

CO2: Apply java concepts like inheritance, polymorphism, interfaces and packages in programming.

CO3: Apply the concepts of Multithreading and Exception handling to develop efficient and error free codes

CO4: Developing web application using advance java concepts like jdbc, servlet and jsp.

CO5: Understanding the concepts of MVC architecture.

Course Contents

UNIT-I

No. of Hours: 8

1. Write a Java program to print 'Hello' on screen and your name on a separate line.
2. Write a Java program to print the sum of two numbers.
3. Write a Java program to divide two numbers and print them on the screen.
4. Write a Java program that takes two numbers as input and displays the product of two numbers.
5. Write a Java program to print the sum (addition), multiply, subtract, divide and remainder of two numbers.
6. Write a Java program to print the area and perimeter of a circle.
7. Write a Java program that takes three numbers as input to calculate and print the average of the numbers.
8. Write a Java program to swap two variables values.
9. Write a Java program to swap two variables values without using third variable.
10. Write a Java program to check whether Java is installed on your computer.
11. Write a Java program to get a number from the user and print whether it is positive or negative.
12. Write a Java program that takes three numbers from the user and prints the greatest number.
13. Write a Java program to find the number of days in a month.
14. Write a Java program that requires the user to enter a single character from the alphabet. Print Vowel or Consonant, depending on user input. If the user input is not a letter (between a and z or A and Z), or is a string of length > 1, print an error message.
15. Write a Java program that takes a year from the user and prints whether it is a leap year or not.
16. Write a program to check whether a number is divisible by 5 and 11 or not.
17. Write a program to check whether a number is even or odd.

18. Write a program to input marks of five subjects Physics, Chemistry, Biology, Mathematics and Computer. Calculate percentage and grade.

19. Write a program to input basic salary of an employee and calculate its Gross salary according to following:

Basic Salary \leq 10000 : HRA = 20%, DA = 80%

Basic Salary \leq 20000 : HRA = 25%, DA = 90%

Basic Salary $>$ 20000 : HRA = 30%, DA = 95%

20. Write a program to input electricity unit charges and calculate total electricity bill according to the given condition:

For first 50 units Rs. 0.50/unit

For next 150 units Rs. 0.75/unit

For next 250 units Rs. 1.20/unit

For unit above 250 Rs. 1.50/unit

An additional surcharge of 20% is added to the bill

21. If his basic salary is less than Rs. 1500, then HRA = 10% of basic salary and DA = 90% of basic salary. If his salary is either equal to or above Rs. 1500, then HRA = Rs. 500 and DA = 98% of basic salary. If the employee's salary is input through the keyboard write a program to find his gross salary.

Write a program to calculate the salary as per the following table

Taken From Original file

UNIT-II

No. of Hours: 8

1. Write a java program to create class called Employee with methods called work() and getSalary(). Create a subclass called HRManager that overrides the work() method and adds a new method called addEmployee().

2. Write a java program to create a class known as Person with methods called getFirstName() and getLastName(). Create a subclass called Employee that adds a new methods named getEmployeeID() and override the getLastName() methods to include the employee's job title.

3. Write a java program to create a class known as "BankAccount" with methods called deposit() and withdraw(). Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdraws if the account balance falls below one hundred.

4. Write a java program to create interface shape with getArea() method. Create three classes Rectangle, Circle and Triangle that implements the shape interface. Implement the getArea() method for each of the three classes.

5. Write a java program to create interface sortable with a method sort() that sorts an array of integers in ascending order. Create two classes Bubblesort and Selectionsort that implement the sortable interface and provide their own implementations of the sort() method.

6. Write a Java program to create an abstract class Animal with an abstract method called sound(). Create subclasses Lion and Tiger that extend the Animal class and implement the sound() method to make a specific sound for each animal.

7. Write a Java program to create an abstract class Employee with abstract methods calculateSalary() and displayInfo(). Create subclasses Manager and Programmer that extend the

Employee class and implement the respective methods to calculate salary and display information for each role

8. Write a Java program to create an abstract class Employee with abstract methods calculateSalary() and displayInfo(). Create subclasses Manager and Programmer that extend the Employee class and implement the respective methods to calculate salary and display information for each role

9. Write a Java program to create a class called "Cat" with instance variables name and age. Implement a default constructor that initializes the name to "Unknown" and the age to 0. Print the values of the variables.

10. Write a Java program to create a class called Student with instance variables studentId, studentName, and grade. Implement a default constructor and a parameterized constructor that takes all three instance variables. Use constructor chaining to initialize the variables. Print the values of the variables

11. Write a Java program to create a basic Java thread that prints "Hello, World!" when executed

12. Write a Java program that performs matrix multiplication using multiple threads.

13. Write a Java program that creates a bank account with concurrent deposits and withdrawals using threads.

14. Write a Java program to create a producer-consumer scenario using the wait() and notify() methods for thread synchronization.

15. Write a Java program to demonstrate Semaphore usage for thread synchronization.

16. Write a simple applet program to display "Hello Applet" Message.

17. Write a applet program to show its life cycle methods.

18. Write a applet program for revolving banner.

19. Connect database to Java program

20. Program to create database table using Java

21. Java Program to insert, update, delete & select records

22. Program to delete record from database

23. Program to execute batch of SQL statements

24. Program to insert date in oracle database

25. Program to insert & read image

26. Program to commit & rollback

27. Program to execute SQL select query

UNIT-III

No. of Hours: 8

1. Write a program to create a servlet that displays the welcome message.

2. Create a simple interest calculator web page using a servlet.

3. Develop a dynamic web application using servlets to create and display cookies.

4. Create a registration form and display the values entered by the user in another page using servlets.

5. Write a program using servlets to display data from a database table.

6. Display a servlet that fetches init parameters from web.xml displaying user information.
7. Write a program using servlets to create and display context parameters.
8. Develop a dynamic web application using servlets to demonstrate doGet(), and doPost() methods.
9. Write an application using servlets to show how forward() method works.
10. Also, demonstrate the use of include() method in servlets.
11. Furthermore, show how sendRedirect() method works using servlets.
12. In order to demonstrate the working of servlet filters create a dynamic web application.

UNIT-IV

No. of Hours: 8

1. Write a program in JSP to display the given text in a paragraph in different font sizes using a loop.
2. Create a dynamic web application in JSP to find the grade according to the marks entered by the user using the switch statement.
3. Build an application in JSP to calculate factorial of a number using recursion.
4. Write a program to create a JSP method that accepts a number and returns its double.
5. Use Page directive in JSP to display the current date.
6. Create a summation method in JSP that accepts an array and display the sum of its element.
7. Write a program to create a JSP page called alert message JSP that displays alert messages if username & password fields are blank and displays welcome user otherwise.
8. Develop an application to demonstrate the include directive in JSP to display the current date from another JSP page.
9. Write a program in JSP to display the properties of an object using the useBean directive.
10. Build an application in JSP that redirects to another page.

UNIT-V

No. of Hours: 8

1. Write java program to demonstrate the functionality of RMI.
2. Write java program to show the use of EJB.
3. Write java program to demonstrate the functionality of MVC.
4. Write java program to show the use of session.

Text Books:

1. Java2:The Complete Reference by HerbertSchildt,TataMcGraw-Hill,8th Edition, 2011.
2. K.Mukhar,“BeginningJavaEE5:From Novice to Professional”, WroxPress.

Reference Books:

1. The Java Programming Language, Ken Arnold, James Gosling, David Holmes, 3rd Edition, Person Education, 2000
2. Head First Java,Kathy, Sierra, Bert Bates, O’Reilly Publication, 2ndEdition,2005

Online Resources:

1. NPTEL Course Video Lectures on "Programming in java" , By Prof. Debasis Samant, IIT Kharagpur
2. Swayam Course Video Lectures on "Web Technology", By Prof. Dr. Ashutosh Kumar Bhatt