SYLLABUS

SCSIT
School of Computer Science & IT
DAVV

MSc (CS/IT)

4th SEMESTER

To produce world-class professionals who have excellent analytical skills, communication skills, team building spirit and ability to work in cross cultural environment.

To produce international quality IT professionals, who can independently design, develop and implement computer applications.

Professionals who dedicate themselves to mankind, who are environment conscious, follow social norms and ethics.

School of Computer Science & IT,
Devi Ahilya Vishwa Vidyalaya, Indore
www.scs.dauniv.ac.in

Course Name: **MSc (CS/IT) 4th Semester**

Subject Code: **CS-5617**

Subject Name: **Internet & Web Technology**

Aim of the Subject

To give students a good understanding of basic concepts of object-oriented program design with the help of real world problem solving using JAVA. Enable students to develop web applications using JSP and Servlet.

Learning Outcomes

The students are expected to learn following after completion of the course:

- basic principles of object-oriented program design using Java.
- advanced issues related to writing classes and methods
- such as data, visibility, scope, method parameters, object references etc.
- Get exposure the ideas behind Inheritance, polymorphism, method overriding, interfaces, packages and exceptions.
- the concept of servers, web technology basics and their challenges. Also
- configuration and testing of Web Server.

Unit 1

Introduction to Java: Features of Java, Object- oriented Programming Overview, Introduction of Java Technologies, JVM architecture and its components, Java Program structure. Arrays, Variable-Length Argument Using Command- line Arguments. Introduction to Class, Objects, Methods and Instance Variables, Naming Conventions, Constructors, Method Overloading, Static Method, Static Field, Math Class, this reference, Garbage collection and finalize. String Handling.

Unit 2

Inheritance: Use of final and Super Keywords Method Overriding, Upcasting, Dynamic Method Dispatch; Packages and Interfaces: Defining a Package, Understanding CLASSPATH, Access Protection, Importing packages, Creating own Packages. Defining an Interface, Properties of Interface, Advantages of Interface, Abstract classes and Abstract Methods;

Unit 3

Exception Handling: Use of try, catch, throw, throws and finally keywords; Multithreading: Introduction, Java Thread Model, Thread priorities, Thread life cycle, Creating Thread, Thread Execution. Streams and Files, Applets: Applet Life Cycle Methods, Executing Applet in Web Browser and in Applet viewer.

Unit 4

Introduction to HTTP, web Server and application Servers, Installation of Application servers, Deployment Descriptors, The Generic Servlet, Lifecycle of Servlet. Servlet Packages, Classes, Interfaces and Methods, Handling Forms with Servlet. Session handling API, Servlet Collaboration, Attributes and various scopes of an Attribute.

Unit 5

JSP Basics: JSP lifecycle, Directives, scripting elements, standard actions, implicit objects, Session handling in JSP, Separating Business Logic and Presentation Logic, Building and using Java Bean. MVC Architecture, Database operations handling in Web applications.

Text Book(s)

1. Java2: The Complete Reference by Herbert Schildt, Tata Mc Graw- Hill, 8th Edition, 2011.

2. M. Hall, L. Brown, " Core Servlets and Java Server Pages", 2 nd edition,Pearson Education.

Reference Material(s)

1. C. Bauer, G. King, "Hibernate in Action", Manning Press

2. B. Basham, K. Sierra, B. Bates, "Head First Servlet and JSP", 2nd Edition, O'Reilly Media.

3. The Java Programming Language, Ken Arnold , James Gosling , David Holmes.

Course Name: **MSc (CS/IT) 4th Semester**

Subject Code: **CS-5216**

Subject Name: **Design and Analysis of Algorithms**

Aim of the Subject

The aim is to teach the basic concepts of algorithms.

Learning Outcomes

The students are expected to learn following after completion of the course:

- Analyze worst-case running times of algorithms using asymptotic analysis.
- Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divid
- Describe the dynamic-programming paradigm and explain when an algorithmic design situation
- calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming
- algorithms, and analyze them.
- Describe the greedy paradigm and explain when an algorithmic design situation calls for it.
- Recite algorithms that employ this paradigm. Synthesize greedy algorithms, and analyze them.
- Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph
- computations as key components, and analyze them.

Unit 1

Recall of asymptotic notation, big-oh, theta, big-omega, and introduce little-oh and little-omega. Worst case and average case complexity

Unit 2

Divide and Conquer: Integer multiplication revisited with an efficient algorithm that motivates and leads into recurrences. Solving recurrences using recurrence trees, repeated substitution, statement of master theorem. Brief recall of merge sort and its recurrence. Median in worst case linear time.

Unit 3

Application of Graph Traversal Techniques: Recall representation of graphs, BFS (as a method for SSSP on unweighted graphs), DFS, connected components, topological sorting of DAGs, biconnected components, and strongly connected components in directed graphs Greedy Algorithms: Greedy choice, optimal substructure property, minimum spanning trees -- Prims and Kruskals, Dijkstra's shortest path using arrays and heaps, fractional knapsack, and Huffman coding (use of priority queue).

Unit 4

Dynamic Programming: Integral knapsack (contrasted with the fractional variant), longest increasing subsequence, Edit distance, matrix chain multiplication, and independent sets in trees. (The instructor may choose a subset that fits within the time frame.)

Unit 5

NP-completeness: reduction amongst problems, classes NP, P, NP-complete, and polynomial time reductions

---

Text Book(s)

[1] Introduction to Algorithms, by Cormen, Leiserson, Rivest, and Stein, MIT Press, Third Edition, 2009. [CLRS]

Reference Material(s)

[1] Algorithms, by Dasgupta, Papadimitrou and Vazirani, McGraw-Hill Education, 2006.

[2] Computer Algorithms, by Horowitz, Sahni, and Rajasekaran, Silicon Press, 2007.