

CS502CT and CS502CP: Object Oriented Programming through Java

Objective:

The course is designed to impart knowledge and skill required to solve real world problems using object-oriented approach utilizing Java language constructs. This course covers the two main parts of Java i.e. Java Language and Java Library

After completion of the course students are expected to understand the following-

- Java tokens for creating expressions and creating data types.
- The way various expressions and data types are assembled in packages.
- Implementation of Inheritance, Exception handling and Multithreading in Java.
- Java I/O basics and Applets.
- Network Programming in Java.
- Accessing relational databases from Java program.
- Java Servlets.

Outline of the Course

Minimum Class Hours			Exam time (Hours)		Marks				
Theory	Practical	Total	Theory	Practical	Theory		Practical		Total
					External	Internal	External	Internal	
60	40	100	2	3	37.5	12.5	37.5	12.5	100

Unit	Topic	Minimum Class Hours			Marks
		Theory	Practical	Total	Theory
I	Introduction to Java Programming, Classes and Methods	12	08	20	7.5
II	Inheritance, Exception handling, Multithreading Enumerations and Autoboxing	12	08	20	7.5
III	Generics, Lambda expressions, String handling, java.lang, java.util and Input/Output	12	08	20	7.5
IV	Applet class, Handling Events and GUI programming with JavaFX	12	08	20	7.5
V	Networking, JDBC, Java Servlets	12	08	20	7.5
Total		60	40	100	37.5

Detailed Syllabus

Unit I: Introduction to Java Programming, Classes and Methods 12+08 Hours

Introduction to Java: Genesis and Overview, Java & Internet, **Object-Oriented Programming features (Abstraction, Encapsulation, Inheritance and Polymorphism)**; Difference between (Java Script and Java, Java and C++, Java applet and Application), Java Development Kit (JDK) Java Virtual Machine (JVM), The Bytecodes, Compile & run a simple program

Constant, Variable, Data types & Arrays: Java Token & Keywords, Primitive Types, Integer literal, Floating point literal, Character literal, Boolean literal, String literal, declaring a variable,

Dynamic initialization, The scope and lifetime of variable, Type conversion and casting, Automatic type promotion in expression, Arrays (One-dimension, Multidimension), Alternative array declaration syntax

Operators: Arithmetic operators, Bitwise operators, Relational operators, Boolean logical operators, The assignment operator, Conditional operator, Operator precedence

Control statement: Decision making and Branching (*if*, Nested *if*, *if-else-if* ladder, *switch*, Nested *switch*, The *?:* operator), Decision making and Looping (*while*, *do-while*, *for*), Jump (*break*, *continue* and *return*)

Introduction to classes, methods and objects: The general form of a class, declaring objects, Assigning object reference variable, Introducing methods (Adding methods to a class, returning a value, Adding methods that takes parameters), Constructors, Parameterized constructor, The *this* keyword, Instance variable hiding, Garbage collection, the *finalize()* method, A stack class-an example, Overloading(methods, constructors), Using object as parameters, Argument passing, Returning objects, Recursion, Introducing Access control (public, private and protected), *static*, *final*, nested and inner classes, String class, Command-line argument, Variable-Length arguments, Scanner (Constructors, Basics, setting delimiters)

Unit II: Inheritance, Exception handling, Multithreading, Enumerations and Autoboxing

12+08 Hours

Inheritance: Extending a class, Basics of Inheritance, Member access and inheritance, using **super**, creating a multilevel hierarchy, when constructors are called, method overriding, dynamic method dispatch, using abstract classes, using *final* with inheritance, the Object class

Packages and Interface: Packages (Defining a package, Finding packages and classpath, Access protection and importing packages), Interfaces (Defining, implementing and Applying Interfaces, Variables in interface, Interfaces can be extended)

Exception handling: Exception handling fundamentals, Exception types (uncaught exceptions, using *try* and *catch*) Nested try statement, multiple *catch* clauses, *throw*, *throws* and *finally*, Java's built-in exceptions, user defined throwable, user defined exception subclasses, using Exception

Multithreaded Programming: The Java thread model (thread priorities, synchronization and inter-thread communication); The main thread, Creating a thread. *isAlive()*, *join()*, Deadlock, suspending, resuming and stopping

Enumerations, Type Wrappers and Autoboxing: Enumeration fundamentals, Java Enumerations are Class Types, Enumerations Inherit Enum, Type Wrappers, autoboxing and methods, autoboxing/unboxing occurs in expressions, autoboxing/unboxing Boolean and Character values

Unit III: Generics, String handling, java.lang, java.util and Input/Output

12+08 Hours

Generics: Introduction, Generic class with Two Type Parameters, General form of a generic class, Bounded Types, Using wildcard arguments, Creating a generic method, Generic Interfaces, Ambiguity errors, Generic Restrictions

java.util package – The Collection Framework – Collection Overview, Collection Interfaces (*Collection*, *List*, *Set*, *SortedSet*), Accessing Collection via an Iterator, Arrays

String handling: The string constructor, Special string operations, Character extraction, String searching & comparison, Data conversion using *valueOf()*, *StringBuffer*

Exploring java.lang: Math functions (transcendental, exponential, rounding)

Input/Output-: Streams (Byte Streams and Character streams- class InputStream, OutputStream, Reader, Writer, Predefined streams, InputStreamReader, BufferedReader, Reading console input, writing console output, Reading and writing files, File, FileNameFilter & Directories, FileInputStream, FileOutputStream, PrintStream, FileReader, FileWriter

Unit IV: Applet Class and Handling Events

12+08 Hours

The applet class: Applet fundamentals, The applet class, Applet architecture, Applet skeleton (initialization and termination, overriding update()), Applet Display Methods, Requesting repainting, Using the Status Window, HTML applet tag, Passing parameters to applets

Handling events:, The Delegation Event Model, Event Classes, Sources of events, Event Listener Interfaces, Processing mouse events, Handling keyboard events, Adapter classes, Anonymous Inner classes

Unit V: Networking, JDBC, Java Servlets

12+08 Hours

Networking: Networking basics, InetAddress, Factory methods, URL, URLConnection, HttpURLConnection, Establishing a simple server using Stream Sockets, Establishing a simple client using Stream Sockets, Connectionless Client/Server Interaction with Datagrams

Java database connectivity (JDBC): Introduction to JDBC, type of JDBC connectivity, Accessing relational database from Java programs, Establishing database connections

Java Servlets: Background, The life cycle of a Servlet, The javax.servlet.http package (HttpServletRequest, HttpServletResponse, HttpSession Interfaces, Cookie class, HttpServlet class, HttpSessionEvent class), Handling Http requests and Response (HTTP GET & HTTP POST), Using Cookies, Session Tracking

Recommended Book

1. **Herbert Schildt**, *Java- The Complete Reference*, McGraw Hill Education (India) Ninth Edition, 2014.

Reference books:

1. **Deitel Deitel**, *Java How to program*, Prentice Hall India Ltd Ninth Edition, 2012.
2. **Kathy Sierra & Bert Bates**, *Head First Java*, O'Reilly Second Edition.
3. **E Balagurusamy**, *Programming with Java A Primer*, McGraw Hill Education Fourth Edition 2006.

Instructions to Paper Setter

Questions should be set according to the following scheme.

Theory Questions				Practical Questions		
Unit	To be set	To be Answered	Marks	To be set	To be Answered	Marks
I	2	1	09	2	1	13.5
II	2	1	09			
III	2	1	09			
IV	2	1	09	3	2	24
V	2	1	09			
Total	10	5	45	5	3	37.5

Practical Assignments

(Questions need not be restricted to this list)

1. Write a program to create a class called Box with a parameterized constructor, along with a method to calculate the volume of the box. Use the class to find the volume of two boxes whose height, width and depth are 10, 20, 30 and 20, 30, 40 respectively.
2. Define a class called stack that can hold 10 integer values, then initialize top of the stack, with push and pop methods. Write a program to push the elements into the stack and pop out from the stack.
3. Write a Java program using a class to multiply two matrixes of 3*3 order. Allow the user to input the values through the keyboard.
4. Write a program to multiply two numbers using a method in a class and pass the values using call by value (pass by value and pass by reference) techniques.
5. Write a Java program to find the factorial of positive integer using recursion.
6. Write a Java program to accept command line arguments and display the arguments along with the positions.
7. Write a Java program to demonstrate method overriding where the program creates a superclass called figure that stores the dimensions of various two-dimensional objects. It also defines a method called area() that computes the area of an object. The program derives two subclasses from figure. The first is Rectangle and the second is Triangle. Each of these subclass overrides area() so that it returns the area of a rectangle and a triangle respectively.
8. Write a Java program to create a thread and start running it using runnable interface. Allow the thread to display a message five times with a gap of 500ms.
9. Write a Java program to demonstrate the synchronization of two threads using the synchronized statement.
10. Write a Java program to demonstrate interthread communication considering the producer and consumer problem. There must be two classes one for producer to produce data and another is consumer to consume data [Hint: Use wait() and notify() to signal in both directions].
11. Write a Java program to copy the contents of one file to another using java.io
12. Write an applet program to accept a message from the keyboard and then to display it on the console.
13. Write an applet to find the biggest of three numbers from the keyboard and display it on the console.
14. Design a Calculator System using Java, The applet should have all the digit buttons along with buttons for operations +, -, *, / and =. There is a designated panel to show the current results. If a digital button is clicked, the number is displayed on the panel. If an operator button is clicked the operation is to be performed. You may assume the expression to be infix. The calculator can operate in two modes
 - i. When the operator buttons are pressed the intermediate results should be displayed
 - ii. The operations can take in any number of arguments and the final result is displayed only when the = button is pressed. [Hint: Use Overloading]

15. Write a program to input integers into an array and sort them using methods. Display the sorted numbers.
16. Write a generic method **printArray** that can print the string representations of the elements of Integer, Double and Character array. [Hint. use **public static < E> void printArray(E[] inputArray)**].
17. Write a generic class **Stack** with one type parameter say T. Provide the necessary constructor and the push() and pop() methods. Test this class with the primitive data types i.e a stack of integer types, stack of double types and stack of character types.
18. Write a program to copy the contents of one file to another file using command line arguments. Give appropriate error messages if any I/O error occurs.
19. Write a socket based Java application program to create a connection between two machines such that whatever text one machine is sending to the other will be displayed at the latter's screen and vice-versa
20. Create a Java application in which a particular machine is configured as the time server which continually listens for requests for time from clients. Clients request the server for time as a result of which the server sends the current time of the clients. The clients make a correction of the received time by adding a very small positive constant to the value and display the corrected time.
21. Create an editor applet in Java using which the users can enter some text and set the font and color of the text according to their choice. The text will be displayed appropriately when the applet is run.
22. Develop various programs that demonstrate the use of the mentioned JavaFX controls.
23. Develop a servlet allowing you to read the names and values of parameters that are included in a client request using ServletRequest class. Develop the web page corresponding to the servlet.
24. Develop a servlet that handles an HTTP GET request. The servlet is involved when a form on a web page is submitted. The HTML web page defines a form that contains a select elements and a submit buttons. The select element name is color and the options are Red, Green and Blue. The servlet responses according to the option submitted and display the message "you have selected color".
25. Develop a servlet that handles an HTTP POST request. The servlet is involved when a form on a web page is submitted. [Hint: The HTML source code is same as the above problem. Except that the method parameter for the form tag explicitly specifies that the POST method should be used and the action parameter for the form tag specifies a different servlet].
26. Write a Java program that prints the addresses and names of the local machine and two well known explored Internet web sites.
27. Implement a simple networked communications client and server. Messages are typed into the window at the server and written across the network to the client side and then they are displayed to demonstrate datagrams.