JAVA ASSIGNMENTS JULY-25

1 message

MBC MANIT BHOPAL <mca523.manit@gmail.com> Fri, 18 Jul, 2025 at 9:04 am
To: MBC MANIT BHOPAL <mca523.manit@gmail.com>

Java THEORY Assignment (SET-B)
(Submit in Hard Copy)

Assignment-1

🔹 1) Basic Concepts, Data Types, Loops, Branching, Arrays, Command Line (Q1–Q2)
Q1. Command-Line Grading System:
Write a Java program that accepts a list of student names and their marks (in one command-line string), parses the input, stores them in an array of objects, and assigns grades based on marks.

Q2. Matrix Operations Using 2D Arrays:
Develop a program that performs addition, subtraction, and multiplication of two matrices using loops and branching.

🔹 2) Class, Objects, Abstract Class, Constructors (Q3–Q4)
Q3. Bank Management System (Object-Oriented):
Create classes for BankAccount, SavingsAccount, and CurrentAccount using inheritance and constructor overloading. Use abstract class Account and include transaction methods with validations.

Q4. Object Pool using Static Constructor & Counter:
Design a system where only 3 objects of a class can be created. Use static variables and constructors to limit object creation.

Assignment-2

🔹 3) Inheritance and Types of Inheritance (Q5–Q6)
Q5. Vehicle Inheritance Hierarchy:
Create a multilevel inheritance structure involving Vehicle → Car → ElectricCar. Add constructors, overridden methods, and demonstrate constructor chaining and access of base class fields.

Q6. Hybrid Inheritance using Interfaces:
Simulate a scenario where Person implements two interfaces Student and Employee. Resolve method conflicts and demonstrate hybrid inheritance.

🔹 4) Polymorphism and Types (Q7–Q8)
Q7. Payment Processing System:
Create a base class Payment and derived classes CreditCard, PayPal, and UPI. Demonstrate runtime polymorphism by calling the processPayment() method from a Payment reference.

Q8. Dynamic Calculation Engine:
Design a Calculator class with method overloading for different data types (int, float, double) and also override a method in a subclass ScientificCalculator.

Assignment-3
🔹 5) Method Overloading & Overriding (Q9–Q10)
Q9. Library System with Overloading:
Design a system where addBook() is overloaded to accept different input formats (ID only, ID + title, ID + title + author). Also override toString() for a custom object display.

Q10. Product Comparison Using Overriding:
Create a superclass Product with a method compare(Product p) that is overridden in subclasses (Electronics, Grocery) to compare based on different parameters.

🔹 6) Interface (Q11–Q12)
Q11. Remote-Control Interface System:
Design an interface RemoteControllable with methods start(), stop(), status(). Implement it in Drone, Robot, and Car. Demonstrate polymorphism with interface references.

Q12. Multiple Interfaces & Conflict Resolution:

Create two interfaces Printable and Scannable both having method process(). Implement both in MultifunctionDevice and resolve method conflict explicitly.

Assignment-4

🔹 7) Package, Access Modifiers (Q13–Q14)
Q13. Custom Package Creation & Access Control:
Create a package utils.math with a class PrimeTools having a protected method isPrime(). Access it in a subclass outside the package using proper modifiers.

Q14. Encapsulation & Access Demo:
Design a system with classes having all 4 access modifiers. Try accessing members from different packages and explain what is accessible and what is not.

🔹 8) Multithreading & Implementations (Q15–Q16)
Q15. Ticket Booking Simulation Using Threads:
Simulate a ticket booking system using threads where multiple users are booking tickets. Ensure synchronization so that no two threads book the same seat.

Q16. Producer-Consumer with Runnable Interface:
Implement the producer-consumer problem using the Runnable interface and thread communication methods (wait(), notify()).

🔹 9) Applet (Q17–Q18)
Q17. Interactive Applet Drawing Tool:
Create an Applet that allows the user to choose shapes (circle, rectangle) from a dropdown and draw them by clicking on the applet area.

Assignment-5

Q18. Applet with Animation:
Design an Applet that animates a bouncing ball. Use thread-based animation inside the applet.

🔹 10) JDBC (Q19–Q20)
Q19. 🗄️ Student Result Management with JDBC:
Design a complete JDBC application with GUI or command-line input to:

Create a results table

Insert marks for students

Calculate and update result grade

Display top 3 students

Q20. JDBC Login System with Exception Handling:
Develop a secure login system using JDBC. Validate user credentials from a users table. Handle SQLExceptions, ClassNotFoundException, and invalid login attempts gracefully.

Q21. Custom Exception – Age Not Valid for Voting
Problem:
Create a custom exception class InvalidAgeException. Write a program that throws this exception if the user age is less than 18 when trying to register for voting.

Q22. File Handling with Exception Catching

Problem:
Write a program that attempts to open and read a text file. Handle FileNotFoundException and IOException appropriately using try-catch-finally.

JAVA LAB Assignment
(Implement & Execute all Problems)

Lab Assignment 1
🔹 1) Basic Concepts, Data Types, Loops, Branching, Array, Command Line Args (Q1–Q10)
Q1. Write a Java program to check whether a number is even or odd using if-else.
Q2. Create a program to take a number from the command line and display its multiplication table using a for loop.
Q3. Write a Java program to find the largest of three numbers using nested if-else.
Q4. Develop a program to reverse a given number using a while loop.
Q5. Write a program to accept n integers in an array and print only even numbers.

Q6. Implement a program to sort an array of integers in ascending order using a loop.
Q7. Write a program to count the number of vowels, consonants, digits, and special characters in a given string.
Q8. Accept an integer array and find the maximum and minimum values.
Q9. Write a program using a switch statement to create a simple calculator.
Q10. Create a Java program to accept and display command-line arguments in reverse order.

Lab Assignment 2

🔹 2) Class, Object, Abstract Class, Constructors (Q11–Q17)
Q11. Define a class Employee with attributes id, name, salary and create multiple objects to display details.
Q12. Create a class with parameterized, default, and copy constructors.
Q13. Implement a class Student that contains a constructor to initialize name, roll number, and grade.
Q14. Demonstrate the use of static variables and methods inside a class.
Q15. Create an abstract class Shape with an abstract method area() and concrete method display(). Extend it in Circle and Rectangle.
Q16. Show the use of this and super keywords with constructors in base and derived classes.
Q17. Create a class Car with constructor overloading and method to display vehicle information.

🔹 3) Inheritance and Types of Inheritance (Q18–Q23)
Q18. Write a program to implement single inheritance using Person → Student.
Q19. Demonstrate multilevel inheritance using classes Animal → Mammal → Human.
Q20. Implement hierarchical inheritance where a base class Shape has subclasses Square, Rectangle, and Triangle.

Lab Assignment 3

Q21. Show how constructor chaining works in multilevel inheritance.
Q22. Explain and demonstrate the limitation of multiple inheritance in Java using classes.
Q23. Write a program to show hybrid inheritance using interface and classes.

🔹 4) Polymorphism & Types of Polymorphism (Q24–Q27)
Q24. Write a program to show compile-time polymorphism (method overloading).
Q25. Write a program to demonstrate runtime polymorphism using Animal and its subclasses with method overriding.
Q26. Show how method resolution occurs in case of overridden methods and dynamic method dispatch.
Q27. Implement a scenario where polymorphism is used to calculate the salary of different types of employees (Manager, Clerk, Salesperson).

🔹 5) Method Overloading & Method Overriding (Q28–Q31)
Q28. Create a class with multiple versions of calculateArea() (for circle, square, rectangle).
Q29. Override the toString() method in a user-defined class Book.
Q30. Show how method overriding works when superclass reference is used to call a subclass method.

Lab Assignment 4

Q31. Demonstrate the difference between method overloading and overriding in a single program.

🔹 6) Interface (Q32–Q34)
Q32. Define an interface Printable and implement it in classes Book and Magazine.
Q33. Create an interface Bank with method getRateOfInterest() and implement it in SBI, ICICI, and Axis.
Q34. Write a program to show how multiple inheritance is achieved using interfaces.

🔹 7) Package, Access Modifiers (Q35–Q37)
Q35. Create a user-defined package utility with a class MathTools that has methods for factorial and prime check.
Q36. Demonstrate the use of all access modifiers (private, default, protected, public) in a package structure.
Q37. Create two packages: pack1 with class A, and pack2 with class B that imports A and uses its members accordingly.

🔹 8) Multithreading & Ways to Implement It (Q38–Q42)
Q38. Write a program to create a thread by extending the Thread class and display current thread info.
Q39. Write a program to implement a thread using the Runnable interface.
Q40. Create two threads: one to print even numbers and another to print odd numbers up to 50.
Q41. Create a program to demonstrate thread synchronization for a banking application (deposit and withdraw).

Lab Assignment 5

Q42. Write a program to implement inter-thread communication using wait() and notify() methods.

🔹 9) Applet (Q43–Q45)
Q43. Create a simple applet that displays “Welcome to Java Applet” using paint() method.
Q44. Create an applet that accepts name and age using <param> tag and displays them on the applet window.
Q45. Write an applet that draws a circle and fills it with a chosen color.

🔹 10) JDBC (Q46–Q50)

Q46. Write a JDBC program to connect with MySQL database and create a table students(id, name, marks).
Q47. Insert 5 records into the students table using JDBC PreparedStatement.
Q48. Retrieve and display all student records where marks > 80.
Q49. Update the name and marks of a student with a specific ID using JDBC.
Q50. Delete a student record from the table using user input via JDBC and show proper exception handling.

Q51. Student Marks Validation (Multiple Custom Exceptions)
Problem:
Design a program that takes student marks input (0–100). Throw NegativeMarksException if marks are negative and MarksOutOfRangeException if marks > 100.

Q52. JDBC Connection Error Handling

Problem:
Simulate a JDBC connection to a non-existent database or with wrong credentials and catch ClassNotFoundException and SQLException. Print detailed error messages to help the user debug.

(Practice Assignment)

🔹 1) Basic Concepts, Data Types, Loops, Branching Statements
Q1. Write a Java program to check whether a given number is prime or not using a loop and if-else.

Q2. Create a Java program to accept 5 integers from the user and print the sum and average using arrays and loops.

🔹 2) Class and Objects, Abstract Class
Q3. Create a class Student with attributes id, name, and marks. Write a program to input and display details of 5 students using objects.

Q4. Create an abstract class Shape with an abstract method area(). Create subclasses Circle and Rectangle to override area() and calculate areas accordingly.

🔹 3) Inheritance
Q5. Create a class Person with fields name and age. Derive a class Employee from Person with additional fields empId and salary. Write a program to initialize and display all details of an employee.

Q6. Implement multilevel inheritance with classes Vehicle → Car → SportsCar and show constructor chaining.

🔹 4) Polymorphism
Q7. Demonstrate compile-time and run-time polymorphism using appropriate examples in Java.

Q8. Create a superclass Animal with method speak(). Create subclasses Dog and Cat which override speak(). Call the method using superclass reference and show polymorphism in action.

🔹 5) Method Overloading and Method Overriding
Q9. Create a class Calculator with overloaded methods add() to add integers, doubles, and 3 numbers.

Q10. Create a base class Bank with method getRateOfInterest(). Override it in subclasses SBI, ICICI, and PNB to return different interest rates.

🔹 6) Interface
Q11. Define an interface Vehicle with methods start() and stop(). Implement this interface in classes Bike and Scooter.

Q12. Create an interface Printable and implement it in classes Book and Magazine. Write a method to print details.

🔹 7) Package
Q13. Create a custom package myutil containing a class MathTools with a method factorial(int n). Import the package in another class and use the method.

Q14. Create a package shapes that contains classes Circle, Square, and Triangle. Each class should have a method area().

🔹 8) Multithreading
Q15. Create a class that extends Thread and prints numbers from 1 to 10 after a delay of 500 ms. Run two threads simultaneously.

Q16. Create a program that demonstrates the use of Runnable interface to implement a thread that prints odd numbers from 1 to 50.

🔹 9) Applet
Q17. Write a simple Java Applet that displays “Welcome to Java Applet!” and override paint() method.

Q18. Create an Applet that takes name and age as parameters and displays them using the paint() method.

- 10) JDBC

Q19. Write a JDBC program to connect to a MySQL database, insert a record into a table students(id, name, marks).

Q20. Write a JDBC program to retrieve and display all records from the students table where marks > 80 using PreparedStatement.

- Q21. ATM Transaction Simulation (Checked Exception)

Problem:
Write a program to simulate an ATM withdrawal. If the user tries to withdraw more money than the balance, throw a java.lang.Exception with the message "Insufficient Balance".

- Q22. Nested Try-Catch: Division and Array Access

Problem:
Write a program that performs division and array access in a nested try-catch block. Handle ArithmeticException and ArrayIndexOutOfBoundsException separately and display meaningful messages.