🪔 Dilwali revision assignment opps concept in java

Topic: Data Types, Wrapper Classes, Operators, and Type Casting

1️⃣ Data Type Basics

👉 Question:

Write a Java program to declare variables of all primitive data types and print their default and assigned values.

2️⃣ Type Casting (Widening)

👉 Question:

Convert an int value to double and print both values before and after conversion.

3️⃣ Type Casting (Narrowing)

👉 Question:

Take a double value and convert it into int. Print the result and explain what happens to the decimal part.

4️⃣ Wrapper Class Conversion

👉 Question:

Convert a primitive int to Integer (autoboxing) and again convert it back to int (unboxing). Print both results.

5️⃣ String to Wrapper Conversion

👉 Question:

Take a String s = "123"; and convert it into an Integer. Then add 10 to it and print the final result.

6️⃣ Arithmetic Operators

👉 Question:

Write a Java program to perform all arithmetic operations (+, -, *, /, %)

between two integers and print the result of each.

7️⃣ Relational & Logical Operators

👉 Question:

Write a program that checks whether a given number is between 10 and 50 using relational and logical operators.

8️⃣ Increment & Decrement

👉 Question:

Demonstrate the difference between ++i and i++ using a small example.

9️⃣ Compound Assignment Operator

👉 Question:

Write a program using compound assignment operators (+=, -=, *=, /=) to manipulate a number and show the final result.

🔟 Mixed Type Expression

👉 Question:

Take an int, float, and double variable, perform an arithmetic operation between them, and observe the final result type. Explain why Java gives that result type.

Topic:

if-else, switch, loops, arrays, 2D array✅✅✅✅✅✅

1️⃣ Write a program to check whether a given number is even or odd using if-else.

2️⃣ Write a program to find the largest of two numbers using if-else.

3️⃣ Write a program to check whether a year is leap year or not.

4️⃣ Write a program to print 1 to 10 using a for loop.

5️⃣ Write a program to calculate the sum of first N natural numbers using a loop.

6️⃣ Write a program to print the multiplication table of any number entered by the user.

7️⃣ Write a program to count the number of digits in a given integer using a while loop.

8️⃣ Write a program that checks whether a character is vowel or consonant using a switch statement.

9️⃣ Write a program to find the factorial of a number using a loop.

🔟 Write a program to reverse an integer number using a while loop.

1️⃣1️⃣ Write a program to find the sum of elements of an array.

1️⃣2️⃣ Write a program to find the maximum and minimum element in an array.

1️⃣3️⃣ Write a program to search a number in an array (Linear Search).

1️⃣4️⃣ Write a program to sort an array in ascending order (using any loop).

1️⃣5️⃣ Write a program to count even and odd elements in an array.

1️⃣6️⃣ Write a program to copy all elements of one array into another

array.

1️⃣7️⃣ Write a program to add two matrices (2D arrays).

1️⃣8️⃣ Write a program to find the transpose of a matrix.

1️⃣9️⃣ Write a program to find the sum of diagonal elements in a square matrix.

2️⃣0️⃣ Write a program to find the element-wise product (multiplication) of two 2D arrays.

Topics: Class, Object, Constructor, Inheritance, Polymorphism, Encapsulation, Abstraction, Interface) ✅✅✅✅✅✅✅✅✅✅

1️⃣ Create a simple class and object

Write a Java class Student with name and age fields, and create an object to display student details.

2️⃣ Constructor example

Create a class Car with a parameterized constructor to initialize

company name and model.

3️⃣ Default Constructor

Create a class Book that uses a default constructor to print “Book object created”.

4️⃣ Method inside class

Create a class Calculator having methods add(), sub(), mul(), div() and call them using an object.

5️⃣ Object reference example

Create two references pointing to the same object and show how changing data in one reference affects the other.

6️⃣ Using this keyword

Create a class Employee with variables name and salary, and initialize

them using this keyword.

7️⃣ Inheritance - Single Level

Create a class Animal and a subclass Dog which extends Animal. Print message using method overriding.

8️⃣ Multilevel Inheritance

Create a class hierarchy: Vehicle → Car → ElectricCar and demonstrate method inheritance.

9️⃣ Method Overloading

Create a class MathOperation having multiple add() methods with different parameters.

🔟 Method Overriding

Create a base class Shape and a subclass Circle which overrides area()

method.

1️⃣1️⃣ Encapsulation Example

Create a class BankAccount with private data members and public getter and setter methods.

1️⃣2️⃣ Abstraction Using Abstract Class

Create an abstract class Shape with an abstract method draw() and subclasses Circle and Rectangle implementing it.

1️⃣3️⃣ Interface Implementation

Create an interface Playable with method play(). Implement it in classes Guitar and Piano.

1️⃣4️⃣ Multiple Interface Implementation

Create two interfaces Printable and Showable and a class Document

that implements both.

1️⃣5️⃣ Upcasting Example

Create classes A and B (B extends A), and show how A obj = new B();

calls overridden methods.

⚙️ B. MEDIUM LEVEL (15 QUESTIONS)

1️⃣6️⃣ Downcasting Example

Demonstrate safe downcasting using instanceof in inheritance

hierarchy.

1️⃣7️⃣ Hierarchical Inheritance

Create classes Employee, Manager, and Developer where both inherit

from Employee. Print their info.

1️⃣8️⃣ Constructor Chaining

Show constructor chaining using super() keyword in parent-child classes.

1️⃣9️⃣ Polymorphism Example

Write a program to demonstrate runtime polymorphism using parent reference and child object.

2️⃣0️⃣ Dynamic Method Dispatch

Demonstrate how Java decides which method to call during runtime (using overridden methods).

2️⃣1️⃣ Abstract Class + Concrete Class Combo

Create an abstract class Account with method calculateInterest(), implemented by subclasses Savings and Current.

2️⃣2️⃣ Encapsulation with Validation

In class User, set private password and validate it using a setter method

(must be >8 chars).

2️⃣3️⃣ Interface with Multiple Implementations

Create interface Payment and implement in CreditCard and UPI

classes. Call common method using interface reference.

2️⃣4️⃣ Real-World Inheritance Example

Create class Person, subclass Teacher, and another subclass Student.

Print their roles using inheritance.

2️⃣5️⃣ Aggregation Example

Create classes Address and Student. Use Address object inside

Student class to demonstrate aggregation.

2️⃣6️⃣ Composition Example

Create class Engine used inside class Car where car cannot exist without engine (tight dependency).

2️⃣7️⃣ Interface Inheritance

Create two interfaces A and B, and another interface C extending both.

Implement C in a class.

2️⃣8️⃣ Polymorphism with Abstract + Interface

Create an abstract class Animal and an interface Pet. Class Dog extends Animal and implements Pet.

2️⃣9️⃣ Final Keyword Example

Create a class with final variable, final method, and final class to demonstrate restrictions.

3️⃣0️⃣ Real-Time Abstraction Example

Design abstract class Database with abstract methods connect() and disconnect(). Implement it for MySQLDatabase and OracleDatabase.