by Kunal Sir

Assignment : Method Overloading (Compile-Time Polymorphism)

Part A: MCQs on Method Overloading

Q1. What is method overloading in Java?

- A. Same method name with same parameters in different classes
- B. Same method name with different parameter lists in the same class
- C. Different method names with same parameters
- D. Same method name and parameters with different return types

---

Q2. Which of the following changes results in valid method overloading?

- A. Changing only return type
- B. Changing method name
- C. Changing number of parameters
- D. Changing access modifier only

Q3. Method overloading is an example of:

- A. Runtime polymorphism
- B. Compile-time polymorphism
- C. Dynamic binding
- D. Method overriding

Q4. Which of the following method signatures are overloaded?

```
void display(int a)
void display(double a)
```

- A. Invalid (same signature)
- B. Valid overloading
- C. Causes runtime error
- D. Causes ambiguity

Q5. Can constructors be overloaded in Java?

- A. No, never
- B. Yes, like methods
- C. Only default constructors
- D. Only parameterized constructors

Q6. Which rule is mandatory for method overloading?

- A. Same return type
- B. Same access modifier
- C. Same method name
- D. Same method body

Q7. What happens if two overloaded methods differ only by return type?

- A. Code compiles successfully
- B. JVM selects correct method
- C. Compile-time error
- D. Runtime exception

Q8. Which of the following is NOT used by the compiler to resolve overloading?

- A. Method name
- B. Number of parameters
- C. Data type of parameters
- D. Return type

Q9. Method overloading improves:

- A. Security
- B. Performance
- C. Code readability
- D. Memory usage

Q10. Which is a correct example of method overloading?

- A. `int sum(int a, int b)` and `double sum(int a, int b)`
- B. `void sum(int a)` and `void sum(int a, int b)`
- C. `void sum(int a)` and `int sum(int a)`
- D. `static void main()` and `static int main()`

Part B: Problem Statements on Method Overloading (with Sample Input & Output)

Problem 1: Create a class `Calculator` with overloaded methods `add()`.

Sample Input:

- add(10, 20)
- add(5, 10, 15)
- add(2.5, 3.5)

Sample Output:

- Sum = 30
- Sum = 30
- Sum = 6.0

Problem 2: Create a class `Shape` with overloaded method `area()`.

Sample Input:

- area(7) // Circle
- area(5, 4) // Rectangle
- area(6, 3, true) // Triangle

Sample Output:

- Area of Circle = 153.94
- Area of Rectangle = 20
- Area of Triangle = 9

Problem 3: Create a class `Printer` with overloaded method `print()`.

Sample Input:

- print(100)
- print("Java")
- print(10, "Programs")

Sample Output:

- 100
- Java
- 10 Programs

Problem 4: Create a class `MathOperation` with overloaded method `multiply()`.

Sample Input:

- multiply(2, 3)
- multiply(2, 3, 4)
- multiply(2.5f, 4.0f)

Sample Output:

- Result = 6
- Result = 24
- Result = 10.0

Problem 5: Create a class `Display` with overloaded method `show()`.

Sample Input:

- show("Amit")
- show("Amit", 20)
- show("Amit", 20, 85.5)

Sample Output:

- Name: Amit
- Name: Amit, Age: 20
- Name: Amit, Age: 20, Marks: 85.5

Problem 6: Create a class `Converter` with overloaded method `convert()`.

Sample Input:

- convert(5)
- convert(200, true)

Sample Output:

- 5000 meters
- 20000 centimeters

Problem 7: Create a class `Salary` with overloaded method `calculateSalary()`.

Sample Input:

- calculateSalary(20000)
- calculateSalary(20000, 5000)

Sample Output:

- Total Salary = 20000
- Total Salary = 25000

Problem 8: Create a class `AreaCalculator` with overloaded method `calculate()`.

Sample Input:

- calculate(4)
- calculate(5, 6)

Sample Output:

- Area of Square = 16
- Area of Rectangle = 30

Problem 9: Create a class `Message` with overloaded method `send()`.

Sample Input:

- send("Hello")
- send("Meeting Today", "High")

Sample Output:

- Message Sent: Hello
- Message Sent: Meeting Today | Priority: High

Problem 10: Create a class `Result` with overloaded method `grade()`.

Sample Input:

- grade(85)
- grade(85, 90)

Sample Output:

- Grade: A
- Grade: A (Attendance Eligible)

Part C: Problem Statements on Constructor Overloading (with Sample Input & Output)

Problem 1: `Student` class with multiple constructors.

Sample Input:

- new Student()
- new Student("Rahul")
- new Student("Rahul", 101)

Sample Output:

- Student Created
- Name: Rahul
- Name: Rahul, Roll No: 101

Problem 2: `Employee` class with overloaded constructors.

Sample Input:

- new Employee(1)
- new Employee(1, "Neha")
- new Employee(1, "Neha", 45000)

Sample Output:

- ID: 1
- ID: 1, Name: Neha
- ID: 1, Name: Neha, Salary: 45000

Problem 3: `Book` class constructor overloading.

Sample Input:

- new Book("Java Basics")
- new Book("Java Basics", "James")
- new Book("Java Basics", "James", 499)

Sample Output:

- Title: Java Basics
- Title: Java Basics, Author: James
- Title: Java Basics, Author: James, Price: 499

Problem 4: `Rectangle` class with overloaded constructors.

Sample Input:

- new Rectangle()
- new Rectangle(5, 10)

Sample Output:

- Area = 0
- Area = 50

Problem 5: `Circle` class constructor overloading.

Sample Input:

- new Circle()
- new Circle(7)

Sample Output:

- Radius = 0
- Area = 153.94

Problem 6: `Car` class constructor overloading.

Sample Input:

- new Car("Toyota")
- new Car("Toyota", "Innova")
- new Car("Toyota", "Innova", 2500000)

Sample Output:

- Brand: Toyota
- Brand: Toyota, Model: Innova
- Brand: Toyota, Model: Innova, Price: 2500000

Problem 7: `BankAccount` class constructor overloading.

Sample Input:

- new BankAccount(12345)
- new BankAccount(12345, "Ravi")
- new BankAccount(12345, "Ravi", 50000)

Sample Output:

- Account No: 12345
- Account No: 12345, Name: Ravi
- Account No: 12345, Name: Ravi, Balance: 50000

Problem 8: `Laptop` class constructor overloading.

Sample Input:

- new Laptop("Dell")
- new Laptop("Dell", 8)
- new Laptop("Dell", 8, 60000)

Sample Output:

- Brand: Dell
- Brand: Dell, RAM: 8GB
- Brand: Dell, RAM: 8GB, Price: 60000

Problem 9: `Product` class constructor overloading.

Sample Input:

- new Product("Pen")
- new Product("Pen", 10)
- new Product("Pen", 10, 100)

Sample Output:

- Product: Pen
- Product: Pen, Quantity: 10
- Product: Pen, Quantity: 10, Price: 100

Problem 10: `Person` class constructor overloading.

Sample Input:

- new Person("Anita")
- new Person("Anita", 22)
- new Person("Anita", 22, "Pune")

Sample Output:

- Name: Anita
- Name: Anita, Age: 22
- Name: Anita, Age: 22, Address: Pune
