

Assignment No. 6:

Page No.	
Date	

Q) What is method overloading? Explain with an example.

Method overloading in Java is a feature that allows a class to have multiple methods with the same name but with different parameters.

This enables the programmer to define several methods with the same name but different behaviors based on the parameters they kept.

When you call an overloaded method, Java determines which method to execute based on the number and type of arguments.

Example:-

```
public class Calculator {
```

```
    public int add (int a, int b) {
        return a + b;
    }
```

```
    public double add (double a, double b) {
        return a + b;
    }
```

```
    public static void main (String [] args) {
```

```
        Calculator calculator = new Calculator ();
        System.out.println (calculator.add (5, 10));
        System.out.println (calculator.add (3.5, 2.5));
    }
```

3

Q) What are the rules for method overloading resolution in Java? How does Java determine which method to call?

In Java, method overloading resolution is determined by the Java compiler based on the following:

- A) Number of parameters
- B) Type of parameters (i.e. int, double)

Example:-

```
public class Test {
```

```
    public void fun (int x, int y) {
        System.out.println ("x = " + x);
        System.out.println ("y = " + y);
    }
}
```

```
public void fun (double x, int y) {
    System.out.println ("x = " + x);
    System.out.println ("y = " + y);
}
```

```
public static void main (String args []) {
```

```
    Test obj = new Test ();
    obj.fun (5, 4);
    obj.fun (3.7, 8);
}
```

```
Two instructions and their outputs are:
```

obj.fun (5, 4); → x = 5
y = 4

obj.fun (3.7, 8); → x = 3.7
y = 8

Two instructions and their outputs are:

obj.fun (3.7, 8); → x = 3.7
y = 8

Two instructions and their outputs are:

3) What does the static keyword mean in Java? Explain the difference between static and non-static method.

= Static keyword is used to define variable that belongs to class itself. When a variable is declared as static it means there will be only one instance of that shared by all instances of the class.

Static Method vs Non-Static Method.

* It belongs to class same. * It belongs to individual class only.

* It can be accessed using class name. * It can access by creating instance of the class.

* Cannot access instance variable directly. To access we have to use "this" word.

4) Can static methods be overloaded and overridden in Java? How are static variables shared across multiple instances of a class?

= Static methods can be overloaded but not overridden in Java.

Static methods cannot be overridden in Java because overriding is

concept related to inheritance and polymorphism, which are based on the dynamic binding mechanism. However static methods are associated with the class itself rather than the instances; so they are resolved at compile time and not subject to dynamic binding.

2) Static Variables in Java are shared across multiple instances of a class because they belong to the class itself, not individual instances. They allocate memory once, shared among all instances, and persist for the entire program. It can be accessed using the class name.

3) What is the role of static keyword in the context of memory management?

4) What is the role of static keyword in Java in the context of memory management? static keyword allocates memory for variables and methods when the class is loaded into memory. static variables are shared amongst all instances of the class, reducing overhead, while static methods are associated with the class itself and can be invoked without creating an instance. This optimization improves memory usage efficiency and facilitates access to shared functionality within the class.

5) What is the role of static keyword in Java in the context of memory management? static keyword facilitates access to shared functionality within the class.

6) What is the significance of "final" keyword in Java? When we use "final" keyword in Java with variable it makes them constant so their values can't be changed or altered.

When we use "final" keyword with a method name it prevents the modification by Subclassing. This ensures that the behaviour of the method remains unchanged.

7) Can a final method be overridden in a Subclass? How does the final keyword affect variables, methods and classes in Java?

In Java, a final method cannot be overridden in a subclass; as it violates the method's implementation. The "final" keyword affects variables by making them immutable. Once initialized, methods by preventing overriding, and classes by disallowing subclassing. This ensures Java prevents unintended modifications and restricts inheritance.

8) What does "this" keyword represent in Java? How is this keyword used in Constructors and methods? In Java, "this" keyword represents a reference to the current object instance within a class.

9) How "this" constructor is used to call another constructor of the same class or to pass the current object instance as an argument to another method.

In methods, "this" is used to access instance variables, invoke other constructors, or refers to the current Object instance explicitly, particularly useful for disambiguating instance variables from local variables or parameters.

9) What are narrowing and widening conversions in Java?

In Java, "narrowing" and "widening" conversions refers to the type conversions that occurs when you assign a value of one data type to a variable of another data type. Transient did assignment.

Widening conversion - When you assign a value of a smaller data type to a variable of larger data type is called as Widening conversion.

ex. int value = 10; float value = 10.0; long value = int value;

Narrowing conversion - When you assign a value of a larger data type to a variable of smaller data type is called as narrowing conversion.

ex. double value = 10.5; int value_1 = double value;

10) Provide examples of narrowing and widening conversions between primitive data types.

Widening conversion :-

ex. `int value = 10;`

`long value1 = int value;`

Narrowing conversion :-

ex. `double value = 10.5;`

`int value2 = double value;`

11) How does Java handles potential loss of precision during narrowing conversions?

Java handles potential loss of precision during narrowing conversions by truncating excess bits without rounding.

When performing such conversions, Java simply discards the extra information beyond the capacity of the target data type. For instance, when casting a `'double'` to an `'int'`, the fractional part is truncated. Developers must be mindful of this behavior to avoid unintended loss of precision when narrowing data types in Java.

12) Explain the concept of automatic widening conversion in Java.

Automatic widening conversion in Java refers to the implicit promotion of smaller data types to a larger data type without explicit casting. It occurs during assignments where smaller data types, like `'int'` or `'float'`, are seamlessly promoted to larger types like `'long'` or `'double'`, ensuring data integrity without loss.

13) What are the implications of narrowing and widening conversions on type compatibility and data loss?

Narrowing and widening conversions impact type compatibility and data loss differently. Widening conversions maintain compatibility by promoting smaller types to a larger types without data loss, ensuring no loss of precision.

However, narrowing conversions require explicit casting and may result in data loss if the source value cannot be accurately represented in the target type, affecting both compatibility and precision. Developers must carefully handle narrowing conversions to prevent unintended loss of data and ensure compatibility between data types.