M T W T F S S

Page No.:

Date:

YOUVA

Assignment-6

1] What is method overloading in java & explain with exa.

→ Method overloading is having two or more methods (or fun's) in class with same name and different arguments / parameters.

Ways to overload the method:

1. By changing number of arguments
2. By changing data type
3. By changing sequence of parameters.

Method overloading is not possible by changing the return type of method only.

```
Class Program{
    void function(double d){ ... }
    void function(int a, int b, double d){ ... }
    void function(int b, int a, double d){ ... }
    psv main(String[] args){ ... }
}
```

3] What does static keyword mean in Java? Explain difference between static and non-static methods.

→ In Java, static keyword means that particular member belongs to type itself, rather than to an instance of that type. This means we'll create only one instance of that static member that's shared across all instances of the class.

Static Methods	Non-static Method
1. Those belongs to the class itself, not to instances.	1. Belong to each instance of class.
2. Accessed using the class name.	2. Accessed using instance of class.
3. Allocate memory only once.	3. Allocate memory for each instance of class.
4. can only directly access static members.	4. can directly access both static and non static members.
5. eg. c`	5. eg.

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

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

2] What are the rules for method overloading resolution in Java? How does Java determine which overloaded method to call?

→ In method overloading, we can create methods having same name but different type, number and sequence of parameters. When an overloaded method is invoked, Java uses type, number and sequence or arguments as its guide to determine which version of overloaded method to call.

3] Can static methods be overloaded and overridden in Java? How are static variables shared across multiple instances of class?

→ In Java, static methods can be overloaded but not overridden. They can have different parameters while having same name in same class or subclass. They cannot be overridden because they act on the class itself, not an object.
Static fields or Class variables. In Java, when we declare a field static, exactly a single copy of that field is created and shared among all instances of that class and can be accessed and modified without creating an instance of class.

5] What is the role of static keyword in the context of memory management?
→ The static keyword in Java is used to share the same variable or method of given class. The size and location of memory blocks are fixed and cannot be changed at runtime.

6] What is the significance of the final keyword in Java?
→ The final keyword is non-access modifier used for classes, attributes and methods, which makes them non-changeable (impo-ssible to inherit or override).
The final keyword is useful when you want a variable to always store the same value.

7] Can a final method be overridden in a subclass? How does the final keyword affect variables, methods and classes in Java?
→ No, the Methods that are declared as final cannot be Overridden.

Final keyword is a non-access modifier used for classes, attributes and methods, which makes them non-changeable. A 'final' class cannot be subclassed, a 'final' method can't be overridden, and a 'final' variable can't be reassigned once initialized.

8] What does the 'this' keywoed represent in Java? How is the this keywoed used in constructors and methods?

→ The 'this' keyword refers to current object in method or constructor. The most common use of this keywoed is to eliminate confusion betn class attributes and parameters with same name because class attributes is shadowed by method or constructor parameter.

9] What are narrowing and widening conversions in Java?

→ Widening conversion is process of converting a small size data type into big size data type.
Narrowing is procer of converting big size data type into small size data type. Also known as explicit conversion.

10] Exa. of narrowing & widening conversion both primitive data types.

→ Narrowing :→ double d = 127.459;
int n = (int) d; → o/p ⇒ 127

Widening: int n = 526;
float f = n; → o/p: f = 526.00

11] How does java handle potential loss of precision during narrowing conversions?

→ Since there is a possibility of data loss, Java requires you to explicitly specify this conversion. This narrowing conversion requires manual operation using the target data type in parentheses.

12] Explain concept of automatic widening conversion in Java.

→ Widening conversion takes place when two data types are automatically converted. This happens when: two data types are compatible, when we assign value of smaller data type to bigger data type.

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

→ Narrowing conversion changes a value to data type that might not be able to hold some of possible values. Widening preserve the source value but can change its representation.