Name - Purva Salvi
PRN - 240340520065

11-3-24

Moodle Assignment - 2
(Java Assignment 6)

Q1. What is method Overloading in Java & explain in example.

→ If a class has multiple methods having same name but different in parameters, it is known as Method Overloading.

• Advantage -
Method Overloading increases the readability of the program.

• Ways to Overload a method -
There are 2 ways to overload method in java:
① By changing number of arguements.
② By changing the data type.

eg ① By changing number of arguements

```
class Adder
{
    static int add( int a, int b) { return a+b; }
    static int add(int a, int b, int c) { return a+b+c; }
}

class TestOverloading1
{
    public static void main(String args[])
    {
        System.out.println (Adder.add (11, 22));
        System.out.println ( Adder.add (11, 22, 33));
    }
}
```

O/P:-
```
33
66
```

eg. ② By changing the datatype of arguements.

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

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

class Overloading2
{
    public static void main( String args[])
    {
        System.out.println( Adder.add (11, 22));
        System.out.println ( Adder.add ( 12.3, 12.6));
    }
}
```

O/P:
```
33
24.9
```

Q2. What are the rules of method Overloading resolution in Java? How does Java determine which method overloaded to call?

→ Key Rules of method Overloading -

① The overloaded & overloading methods must be in the same class. (This includes any methods inherited, even implicitly, from a superclass.)

② The method parameters must change: either the number of parameters or the type of parameters must be different in the two methods.

The compiler distinguishes overloaded methods by their signatures - a combination of the methods name and the number, types and order of its parameters, but not its return type.

Q3. What does the static keyword mean in Java? Explain the difference between static & non-static methods.

→ Static keyword in java indicates that a particular member is not an instance, but rather a part of type. The static member will be shared among all instances of the class, so we will only create one instance of it.
The static keyword is mainly used for memory management. The static keyword in java is used to share the same variable or method of a given class. The static keyword belongs to the class, than an instance. The static keyword is used for a constant variable or method that is the same for every instance of a class. The static keyword is a non-access modifier & that is applicable for the following:
(i) Blocks (ii) Variables (iii) methods (iv) Class / Nested classes.

Difference -
A static method is a class method and belongs to the class itself. This means you do not need any instance in order to use a static method.
A non-static method is an instance method and belongs to each object that is generated from the class.

Q4. Can static methods be overloaded & overridden in Java? How are static variables shared across multiple instances of a class?

→ In Java, static methods can be overloaded but not overridden. They can have diff parameters while having the same name in the same class or subclass.
They cannot be overridden because they act on the class itself, not an object.

The 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 the instances of that class.

Q5. What is the role of static keyword in the context of memory management?

→ There are several benefits of using the 'static' keyword. It helps in memory management as static variables are shared among all instances, reducing the amount of memory required.

Few charecteristics of 'static' keyword-
① Shared memory allocation-
Static variables & methods are allocated memory space only once during the execution of the program.

② Accessible without object instantiation-
Static members can be accessed without the need to create an instance of the class.

③ Associated with class, not objects -
This means that changes to a static member are reflected in all instances of a class, and that you can access static members using the classname rather than object reference.

④ Cannot access non-static members -
static methods and variables cannot access non-static member of a class, as they are not associated with any particular instance of the class.

⑤ Can be overloaded, but not overridden -
Static methods can be overloaded, which means that you can define multiple methods with same name but diff parameters. However, they cannot be overridden, as they are associated with the class rather than with a particular instance of the class.

Q6. What is the significance of the final keyword in Java?
→ The final keyword is a non-access modifier used for classes, attributes & methods, which makes them non-changeable. (impossible to inherit or override).
The final keyword is called as 'modifier'.
The final keyword is useful when you want a variable to always store the same value.

Q7. Can a final method be overridden in a subclass? How does the final keyword affect variables, methods & classes in Java?
→ No, the methods that are declared as final cannot be overridden or hidden.

In Java, the keyword 'final' serves as a non-access modifier applicable to classes, methods & variables.
A 'final' class cannot be sub-classed.
A 'final' method cannot be overridden.
A 'final' variable cannot be reassigned once initialized.

Q8. What does the 'this' keyword represent in java?
How is this keyword used in constructors & methods?
→ The keyword 'this' in java serves a fundamental purpose: it refers to the current object. In other words, 'this' represents the instance of the class where its used. It is commonly used to access or modify the fields of the current object. especially when field names are the same as local variable names.

Within an instance method or constructor, this is a reference to the current object. The object whose method or constructor is being called. You can refer to any member of the current object from within an instance method or constructor by using this.

Q9. What are the narrowing & widening conversions in Java?
→ Widening conversion occurs when a value of one type is converted to another type that is of equal or greater size. Narrowing conversion occurs when a value of one type is converted to a value of another type that is of a smaller size.

Q10. Provide examples of narrowing & widening conversions between primitive data types.

→

Narrow Type casting →

$$\begin{tikzcd}\boxed{\text{double}} \arrow[r] & \boxed{\text{float}} \arrow[r] & \boxed{\text{long}} \arrow[r] & \boxed{\text{int}} \arrow[r] & \boxed{\text{short}} \arrow[r] & \boxed{\text{byte}}\end{tikzcd}$$

←

Widening Typecasting

Example of Widening :-

```
int x=7 ;
long y = x;
 float z = y ;
O/p :-  x =7  ,   y = 7  ,  z = 7.0
```

Example of Narrowing –

```
 double d = 166.60;
   long l = double (d);
    int  i = int (l);
O/p :-   d = 166.60,  l = 166 ,  i = 166
```

Q11. How does Java handle potential loss of precision during narrowing conversions?

→ The easy way of converting primitives to avoid lossy conversion is through downcasting. In other words, casting the larged-sized type to a smaller sized-type. Hence, it is also called as narrowing primitive conversion.

Q12. Explain the concept of automatic widening conversion in Java.

→ Widening conversion takes place when two data types are automatically converted. This happens when:
(i) Two datatypes are compatible. When we assign a value of a smaller data type to bigger datatype.

Q13. What are the implications of narrowing and widening conversions on type compatibility and data loss?

→ Widening conversions are generally safer than narrowing conversions because they usually don't cause data loss. Widening conversions change a value to a data type that that can have any possible value of the original data, while narrowing conversions change a value to a data type that might not be able to hold some of the possible values.
eg. Converting an integral data type to decimal, or from char to String, is a widening conversions.
However, a fractional value is rounded when it is converted to an integral type, and a numeric type being converted to boolean is either reduced to true or false, which is narrowing conversion.

—— x —— x ——