

## Assignment No - 26

Q1) What is difference between inheritance in C++ and inheritance in Java.

→

C++

- Supports Single, Multiple Hierarchical and Hybrid inheritance.

② A class can inherit from multiple classes

③ When a class inherits from two classes that have the same base class. solved using virtual inheritance

④ Class Derived : public Base {}

⑤ Base class members are private by default and not inherited

⑥ Uses virtual functions to achieve runtime polymorphism

⑦ Supports abstract classes using pure virtual function

→ Java

Supports Single, Multilevel and Hierarchical Inheritance (Multiple & Hybrid also)

Avoid multiple inheritance due to ambiguity, use interface

Diamond problem does not occur since java does not allow multiple class inheritance instead, it uses interfaces to achieve similar

class Derived extends Base {}

Default access modifier allows access within the same package.

Uses method overriding and all non final methods are virtual by default

Support abstract classes using abstract keyword

Q) What are different types of inheritance in java?  
Inheritance in java allows a class to acquire properties and behaviors from another class.  
Java supports 3 types of inheritance.

### 1) Single Inheritance

In it there is only one parent and child relationship.  
In this inheritance base class extends by derived class.

Base

↑ extends

Derived

### 2) Multi-level Inheritance

To achieve multi-level inheritance we need more than one class in which there is relationship of parent and child.

At least three classes are required.

Grand parent

↑ extends

parent

↑ extends

child

### 3) Hierarchical Inheritance

It is not standard type of inheritance.

In this multiple classes gets inherited from single or one class.

college Parent

Computer dept

E&TC dept

IT dept child

(Q3) Explain the concept of this keyword in java

- The this keyword in java is a reference variable that refers to the current instance of class. It is used to eliminate ambiguity between instance variables and parameters, enable constructor chaining, and return the current object.

- Uses of this

- 1] Referring to instance variables

- When local variables and instance variables have the same name, this differentiate them.

- ex- this.variableName

- 2] Calling another constructor in the same class (this)

- 3] Returning the current class instance

- 4] Passing the current object as an argument

- 5] Calling methods of the current object

- 6] Using this with inner classes

(Q4) Explain the concept of super keyword in java

- Super is not available in C, C++

- Super keyword is most important if we have to access the parent class characteristics and behavior directly.

- Super class used for

- 1] To access characteristics for parent class

- 2] To access behaviour of parent class

- 3] To access the constructor of parent class

(Q5) How we can call one constructor from the another constructor of same class.

- In java, we can call one constructor from another constructor within the same class using the this() keyword. This is known as constructor chaining.

and helps in code reuse and reducing redundancy.

• Rules for using `this()` to call another constructor.

1] `this()` must be the first statement inside the constructor.

2] It helps in avoiding duplicate code by reusing other constructors.

3] It cannot be used inside static methods.

Q. How we call one constructor from the another constructor of derived class.

→ In Java, we use the `super()` keyword to call a constructor of the parent class from the constructor of the child class.

This process is known as constructor chaining in inheritance and ensures that the parent class is properly initialized before the child class.

• Rules for using `super()` in constructor chaining.

1] `super()` must be the first statement inside the child constructor.

2] If a parent class has a parameterized constructor, the child class must explicitly call `super(parameters)` because Java does not automatically call parameterized constructors.

3] If a parent class has only a default constructor, the compiler automatically inserts `super();`.

Q. Explain the scenario in which we have to call the base class constructor of base class from the derived class explicitly.

→ In Java, when an object of derived class is created, the constructor of the base class is executed first.

This ensures that the parent class is properly initialised.

before child class.

By default, java automatically calls the ~~defoult~~  
constructor of the parent class.

However in some cases, we must explicitly  
call the base class constructor using the super() to  
when do we need to explicitly call the base  
class constructor.

1) when the base class has a parameterized  
constructor.

2) when we need to pass specific arguments to  
the base class constructor.

3) when the base class performs important initializations.

4) when the base class has multiple constructors.

Q) Write a java program which accept one word from user through command line and display length of that word.

→ import java.util.Scanner;

public class length {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a word:");

String word = scanner.next();

System.out.println("The length of the word "+  
word + " is: " + word.length());

scanner.close();

}

Q) Write a java program which accept n words from user through command line and display the length of largest word.

```
→ import java.util.Scanner;
```

```
public class LargestWordLengthScanner {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.println("Enter words");  
        String input = scanner.nextLine();  
        String[] words = input.split(" + ");
```

```
        String largestWord = "";
```

```
        int maxLength = 0;
```

```
        for (String word : words) {
```

```
            maxLength = 0;
```

```
            for (int i = 0; i < word.length(); i++) {
```

```
                if (word.charAt(i) > maxLength) {
```

```
                    maxLength = word.length();
```

```
                    largestWord = word;
```

```
}
```

```
}
```

```
        System.out.println("The largest word is :" + "
```

```
        largestWord + " with length :" + maxLength);
```

```
        scanner.close();
```

```
}
```

```
}
```

Output : Enter words :

vaishnavi Jadhav typsccds

The largest word is : "vaishnavi" with length : 9

Q) What is the use of classpath for java program?

- The classpath in java is an environment variable or command-line option that tells the java virtual machine and compiler where to find user-defined classes, packages and external libraries.
- By default, java looks for classes in the current working directory but when additional class files or JARs are needed, classpath helps locate them.

• Uses -

- 1) Locating class files
- 2) Using external libraries (JAR files)
- 3) Supporting Package structures
- 4) Running Java programs from different directories
- 5) Avoiding 'ClassNotFoundException' and 'NoClassDefFoundError'

↳ Java - beans - good

↳ Java - beans - bad

↳ Java - beans - good

↳ Java - beans - bad

↳ Java - beans - good

↳ Java - beans - bad