

Yash Mahajan SE IT B 04

A] Extended Theory

- 1) State any four points of differentiation between compile time and runtime polymorphism.

Compile-time Polymorphism	Run-time Polymorphism
1) Compile time polymorphism means binding is occurring at compile time.	1) Runtime polymorphism means binding is occurring at runtime.
2) In compile time polymorphism call is resolved by the compiler.	2) In runtime, call is not resolved by the compiler.
3) It is achieved by method and operator overloading.	3) It is achieved by method overriding and and and virtual functions.
4) It provides fast execution because it is known early at compile time.	4) It provides slow execution because it is known at runtime.
5) It is less flexible.	5) It is more flexible.

Yash Mahajan SE IT B 04

2) Explain use of Super keyword with syntax & example.

→ The super keyword refers to superclass (parent) objects.

→ It is used to call superclass methods & to access the superclass constructor.

→ The most common use of the super keyword is to eliminate the confusion between superclass & sub-class that have methods with same name.

Eg) When we have the data members of the same name in both the base & derived class, we can use the super keyword to access the member of base class, data in derived class.

class Base {

 int num = 30;

class Derived extends Base {

 int num = 20;

 void callThis() {

 System.out.println("Base num: " + super.num);

 System.out.println("Derived num: " + num);

 }

class Test {

 public static void main(String[] args) {

 Derived obj = new Derived();

 obj.callThis();

}

Dash Mahajan SE IT B Ok

Output :-

Base num = 30

Derived num = 20

Eg. Invoking Base class constructor

The super keyword can also be used to invoke the parent class constructor, both parameterized & non parameterized in derived class.

// Base Class

class Base {

 Base () {

 System.out.println("Base class constructor");

}

// Derived Class

class ~~derived~~ Derived extends Base {

 Derived () {

 super();

 System.out.println("Derived class constructor");

}

}

class Test {

 public static void main (String args[]) {

 Derived d = new Derived();

}

}

Output:-

Base class constructor

Derived class constructor

Yash Mahajan SE IT B 04

Q] Conclusion :-

In this experiment we have implemented the Object Oriented Programming concept of Inheritance. We have written programs to study at how it is achieved and the what are its various types.

The aim of inheritance is to provide reusability of code so that repetition can be avoided.

Inheritance mainly has two important uses, first is to reuse code to avoid repetition of code and overriding which allows a sub class to provide specific implementation of code that already exists.