6TH MAY TASK

1. Constructor invocation in inheritance hierarchy

Invocation of constructors depends on the type of inheritance being implemented.

Example: Single Inheritance

class A {

A() {

System.out.println("Hi");

}

}

class B extends A {

B() {

System.out.println("Hello");

}

}

public class Sample {

public static void main(String[] args) {

B b = new b();

}

}

Output:

Hi

Hello

1. Difference between no argument constructor and default constructor

Every class must have at least one constructor. If not declared explicitly, java compiler provides a non-parameterised constructor, i.e, default constructor. This default constructor calls its parent class’s non-parameterised constructor. It initializes class variables to their default values.

Example:

Class A {}

Class B extends A {} // Default constructor of class "B" will automatically calls non-arg constructor of class "A" and intialzes value to the variables

While non-arg constructor is defined by a programmer only. It can also intialize the variables. One more point to add here is that super() call is automatically added by the java compiler, if doesn’t find super() in derived class.

Example:

Class A {

int x;  
public A() {  
}  
public int getX() {  
return x;  
}

}

public class B extends A {  
private int y;  
  
public B() { //super() will be automatically used  
}  
  
public int getY() {  
return y;  
}  
  
public void setY(int y) {  
this.y = y;  
}  
}

B b = new b();  
System.*out*.println("A x value => " + d.getX());

System.*out*.println("A y value => " + d.getY());

Result:

A x value => 0

A y value => 0

3.If my base class method is static can i override that method in sub class?

No, we can’t override the static method of parent class in sub class.

The actual flavor of overriding is linked with an object. Overriding is when we pass a child class object to a parent class reference variable but the static methods are associated with class they are not linked with an object so we can't override a static method but we can hide a static method in java.

4.

class Employee{

Public Employee(){

System.out.println(“Emp”);

}

}

class Manager Extends Employee{

Manager(){

System.out.println(“Manager”);

}

}

class Admin Extends Manager{

Admin(){

System.out.println(“Admin”);

}

}

class Sample{

Public static void main(String args[]){

Admin A1=new A1();

System.out.println(“A1”);

}

}

5.a)tv, dvd, laptop if these are subclasses what is the base class?

Electronics will be the base class

b)Implement dynamic polymorphism

class Electronics{

E(){

System.out.println(“Electronic device”);

}

}

class Tv Extends Electronics{

Sound(){

System.out.println(“Tv Sound”);

}

}

class Dvd Extends Electronics{

Sound(){

System.out.println(“Dvd Sound”);

}

}

class Laptop{

Sound(){

System.out.println(“Laptop Sound”);

}

}

Class Sample{

Public static void main(String args[])

{

Tv t1=new t1();

Dvd d1=new d1();

Laptop l1=new l1();

}

}