**JAVA ASSIGNMENT 2**

1)Write Examples of all types of Inheritances?

A) **Inheritance in Java** is a mechanism in which one object acquires all the properties and behaviours of a parent object. It is an important part of  Object Oriented programming system. The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

**Why We Use Inheritance In Java:**

* For Method Overriding
* For Code Reusability

**Types of Inheritance:**

* Single Inheritance.
* Multilevel Inheritance.
* Hierarchical Inheritance.

1. Single Inheritance:

When a class inherits another class, it is known as a single inheritance. In the example given below, Dog class inherits the Animal class, so there is the single inheritance.

Example:

**class** Animal{

**void** eat()

{

System.out.println("eating...");

}

}

**class** Dog **extends** Animal

{

**void** bark(){

System.out.println("barking...");

}

}

**class** TestInheritance{

**public** **static** **void** main(String args[]){

Dog d=**new** Dog();

d.bark();

d.eat();

}

}

1. Multilevel Inheritance

When there is a chain of inheritance, it is known as multilevel inheritance. As you can see in the example given below, BabyDog class inherits the Dog class which again inherits the Animal class, so there is a multilevel inheritance.

Example:

**class** Animal

{

**void** eat(){

System.out.println("eating...");

}

}

**class** Dog **extends** Animal{

**void** bark(){

System.out.println("barking...");

}

}

**class** BabyDog **extends** Dog{

**void** weep(){

System.out.println("weeping...");

}

}

**class** TestInheritance2{

**public** **static** **void** main(String args[]){

BabyDog d=**new** BabyDog();

d.weep();

d.bark();

d.eat();

}

}

1. Hierarchical Inheritance:

When two or more classes inherits a single class, it is known as hierarchical inheritance. In the example given below, Dog and Cat classes inherits the Animal class, so there is hierarchical inheritance.

Example:

**class** Animal{

**void** eat(){

System.out.println("eating...");

}

}

**class** Dog **extends** Animal{

**void** bark(){

System.out.println("barking...");

}

}

**class** Cat **extends** Animal{

**void** meow(){

System.out.println("meowing...");

}

}

**class** TestInheritance3{

**public** **static** **void** main(String args[]){

Cat c=**new** Cat();

c.meow();

c.eat();

}

}

2)Class,Parent class and all access Modifiers?

A)  public class RawProducts{

Private int product id;

Protected String productname;

Public int GetProductId()

{

return productid;

}

Public void SetProductId(int prodid)

{

Productid=prodid;

}

Public String GetProductName()

{

Return productname;

}

Publc void SetProductName(String prodname)

{

Productname=prodname;

}

Public class Produscts extends Rawproducts

{

Public static void main(String[] args)

{

RawProducts r=new RawProducts();

r.SerProductid(1);

System.out.println(r.GetProductId());

r.setProductname(“kesari”);

System.out.println(r.GetProductname());

}

}