Java Assignment-2

1.Parent Class with all access specifiers and getter, setter Methods?

```
package inheritance;

public class Rawproducts {

	private int productid;
	protected String productname;
	public Double price=100.00;
public int GetProductId()
{
	return productid;
}
public void SetProductid(int prodid)
{
	productid=prodid;
}
public String GetProductName()
{
	return productname;
}
public void SetProductName(String prodname)
{
	productname=prodname;
}
public void ProdPrice()
{
	System.out.println("Product Price is "+price);
}
}

public class Products extends Rawproducts{

	public static void main(String[] args) {
		Rawproducts r=new Rawproducts();
		r.SetProductid(1);
		System.out.println("Product Id: "+r.GetProductId());
		r.SetProductName("Moong Dal");
		System.out.println("ProductName: "+r.GetProductName());
		r.ProdPrice();
	}

}
```

Output:

```
Product Id: 1
ProductName: Moong Dal
Product Price is 100.0
```

2. Single Inheritance:

Ex:

```
public class Person {
 public int age,id;
 public String name;
 void naming(String name)
 {
       System.out.println("Name:"+name);
 }
}
public class Student extends Person {
       private char marks='S';
    void ageN (int age)
    {
       System.out.println("Age of Student is:"+age);
    }
    public char Getmarks()
    {
       return marks;
    }

   public void Setmarks(char grade)
   {
          marks=grade;
   }

}

public class Single {

       public static void main(String[] args) {
              Student s=new Student();
              s.naming("Sowjanya");
              s.ageN(21);
              char Marks=s.Getmarks();
              System.out.println("Grade is:"+Marks);
              s.Setmarks(Marks);
       }

}
```

Output:

```
Name:Sowjanya
Age of Student is:21
Grade is:S
```

Multi-Level Inheritance:

Ex:

```
public class Project {
```

```
     public int projectid=101;
     public String projectName="Driver Drowsiness Project";
     public void display()
     {
        System.out.println("Project ID is:"+projectid);
        System.out.println("Project Name is:"+projectName);
     }

}

public class Module extends Project {
   private String Module="ML";
   public String GetModule()
   {
       return Module;
   }

  public void SetModule(String mod)
  {
         Module=mod;
  }
}

public class Task extends Module {
     protected String Taskname="Introduction";
     public String GetTaskname()
     {
       return Taskname;
     }
    public void SetTaskname(String tname)
    {
         Taskname=tname;
    }
}

public class MultiLevel {

       public static void main(String[] args) {
              Project p=new Project();
              p.display();
              Module m=new Module();
              String mod=m.GetModule();
              System.out.println("Project Module is:"+mod);
              m.SetModule(mod);
              Task t=new Task();
              String tname=t.GetTaskname();
              System.out.println("Task is:"+tname);
              t.SetTaskname(tname);
       }
}
```

Output:

```
Project ID is:101
Project Name is:Driver Drowsiness Project
Project Module is:ML
Task is:Introduction
```

Hierarchical Inheritance:

Ex:

```
 public class Picture {
   public String title;
   public double price;
   public void updatePrice(String title,double price)
   {
          System.out.println("Price is:"+price);
          System.out.println("Title Is:"+title);
   }
}
public class Photograph extends Picture{
     public String photographer="Alex";
     protected String camera="Nikon";
     public void alterContrast()
     {
        System.out.println("Photographer name is:"+photographer);
        System.out.println("camera is:"+camera);
     }
}
public class Painting extends Picture {
     public String artist="Sowjanya";
     private String type="Oil Painting";
     protected String owner="John";
     public void printProvenance()
     {
        System.out.println("Artist:"+artist);
        System.out.println("Type:"+type);
        System.out.println("Owner:"+owner);
     }
}
public class Hierarchy {

       public static void main(String[] args) {
              Picture pic=new Picture();
              pic.updatePrice("Flowers",450.00);
              Photograph ph=new Photograph();
              ph.alterContrast();
              Painting pt=new Painting();
              pt.printProvenance();

       }

}
```

Output:

```
Price is:450.0
Title Is:Flowers
Photographer name is:Alex
camera is:Nikon
Artist:Sowjanya
Type:Oil Painting
```

```
Owner:John
```

Multiple Inheritance:

Ex:

```
package inheritance;
interface MotorBike
{
int speed=50;
public void totalDistance();
}
interface Cycle
{
int distance=150;
public void speed();
}
public class TwoWheeler implements MotorBike,Cycle {
	int totalDistance;
	int avgSpeed;
	public void totalDistance()
	{
	totalDistance=speed*distance;
	System.out.println("Total Distance Travelled : "+totalDistance);
	}
	public void speed()
	{
	int avgSpeed=totalDistance/speed;
	System.out.println("Average Speed maintained : "+avgSpeed);
	}
	public static void main(String[] args) {
		TwoWheeler t1=new TwoWheeler();
		t1.totalDistance();
		t1.speed();

	}

}
```

Abstract Class:

Ex:

```
abstract class Bank{
                    abstract int getRateOfInterest();
                    }
                    class SBI extends Bank{
                    int getRateOfInterest()
                    {
                    return 7;
                    }
                    }
                    class AXIS extends Bank{
                    int getRateOfInterest()
                    {
                    return 8;
                    }
                    }
```

```
public class TestBank {

	public static void main(String[] args) {
		Bank b;
		b=new SBI();
		System.out.println("Rate of Interest is: "+b.getRateOfInterest()+"
%");
		b=new AXIS();
		System.out.println("Rate of Interest is: "+b.getRateOfInterest()+"
%");

	}

}
```

Output:

```
Rate of Interest is: 7 %
Rate of Interest is: 8 %
```