1. **package** org.virt.com;

**public** **class** Assignment1

{

**private** **int** rollno;

**protected** String name;

//getters and setters

**public** **int** getRollno()

{

**return** rollno;

}

**public** **void** setRollno(**int** rollno)

{

**this**.rollno = rollno;

}

**public** String getName()

{

**return** name;

}

**public** **void** setName(String name)

{

**this**.name=name;;

}

}

**public** **class** AssignmentMain **extends** Assignment1

{

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

{

Assignment1 a1=**new** Assignment1();

a1.setRollno(3);

System.***out***.println(a1.getRollno());

a1.setName("poloju");

System.***out***.println(a1.getName());

}

}

2.a)SINGLE:

1. **class** Animal{
2. **void** eat(){System.out.println("eating...");}
3. }
4. **class** Dog **extends** Animal{
5. **void** bark(){System.out.println("barking...");}
6. }
7. **class** TestInheritance{
8. **public** **static** **void** main(String args[]){
9. Dog d=**new** Dog();
10. d.bark();
11. d.eat();
12. }}

Output:

barking...

eating...

B)MULTILEVEL

1. **class** Animal{
2. **void** eat(){System.out.println("eating...");}
3. }
4. **class** Dog **extends** Animal{
5. **void** bark(){System.out.println("barking...");}
6. }
7. **class** BabyDog **extends** Dog{
8. **void** weep(){System.out.println("weeping...");}
9. }
10. **class** TestInheritance2{
11. **public** **static** **void** main(String args[]){
12. BabyDog d=**new** BabyDog();
13. d.weep();
14. d.bark();
15. d.eat();
16. }}

Output:

weeping...

barking...

eating...

C)HIRARICAL

1. **class** Animal{
2. **void** eat(){System.out.println("eating...");}
3. }
4. **class** Dog **extends** Animal{
5. **void** bark(){System.out.println("barking...");}
6. }
7. **class** Cat **extends** Animal{
8. **void** meow(){System.out.println("meowing...");}
9. }
10. **class** TestInheritance3{
11. **public** **static** **void** main(String args[]){
12. Cat c=**new** Cat();
13. c.meow();
14. c.eat();
15. }}

Output:

meowing...

eating...

D}MULTIPLE

1. **class** A{
2. **void** msg(){System.out.println("Hello");}
3. }
4. **class** B{
5. **void** msg(){System.out.println("Welcome");}
6. }
7. **class** C **extends** A,B
8. {
9. **public** **static** **void** main(String args[]){
10. C obj=**new** C();
11. obj.msg();
12. }
13. }

Compile Time Error

E)HYBRID with interfaces

public class ClassA

{

public void dispA()

{

System.out.println("disp method of ClassA");

}

}

public interface InterfaceB

{

public void show();

}

public interface InterfaceC

{

public void show();

}

public class ClassD extends ClassA implements InterfaceB,InterfaceC

{

public void show()

{

System.out.println("show method implementation");

}

public void dispD()

{

System.out.println("disp method of ClassD");

}

public static void main(String args[])

{

ClassD d = new ClassD();

d.dispD();

d.show();

}

}

**Output :**

disp method of ClassD

show method implementation