Assignment-3

1)Instance methods in patient calss are

class Patient

* createProfile();
* checkDisease();
* provideTreatment();
* provideMedicine();

2)Public instance members

class CPU

* pendrive
* CD
* external\_keyboard
* external\_mouse
* powersupply

Private instance members

* os
* Cache\_memory
* ram
* alu
* bus
* registers

3)static members of Media class

Class Media

* television
* radio
* news\_paper

In all these types of media once the content is printed or broadcasted they can’t be changed ,even if changes are required in them.Hence they are static members.

4) class Calculator

{

void mul()

{

Int a=10,int b=20;

Int c=a\*b;

System.out.println(“multiplication result is :”,c);

}

class Standard extends calculator

{

void mul(int a ,int b)

{

Int c=a\*b;

System.out.println(“multiplication result is :”,c);

}

)class Scientific extends Calculator

{

void mul()

{

Int a=10,int b=20;

System.out.println(“multiplication result is :”, a\*b);

}

}

Class Math

{

Public static void main(String[] args)

{

Calculator cal=new Calculator();

Standard stan=new Standard();

Scientific sci=new Scientific();

cal.mul();

stan.mul(3,8);

sci.mul();

}

If child class has the same method as declared in the parent class, it is known as **method overriding in Java**. In other words, If a subclass provides the specific implementation of the method that has been declared by one of its parent class, it is known as method overriding. In the above example parent class is “calculator” and child classes are “scientific” and “standard’. The parent and child class have the same method that is “mul”. But implementation or the body of “mul” method in parent child class is diffrent.

5)

interface Flying

{

public void fly()

}

class Bird implements Flying

{

public void fly()

{

System.out.println(“ Bird can fly”);

System.out.println(“ Bird fly range is less as compared to others”);

}

class Superman implements Flying

{

public void fly()

{

System.out.println(“ supper man can fly”);

}

class AeroPlane implements Flying

{

public void fly()

{

System.out.println(“ Aeroplan can fly”);

}

class Messile implements Flying

{

public void fly()

{

System.out.println(“ Messile can fly”);

System.out.println(“ speed is more compared to others”);

}

In Java, there are not only classes but also interfaces. These are introduced with the keyword interface instead of the keyword class. Interface in java, like classes, should each be stored in a separate Java file. In terms of structure, interfaces are very similar to a class; they are almost comparable to abstract classes that only contain method declarations. The only difference to a class is that an interface in java has no implementations, just method headers, and constants

Here interface Fly is common to all classes that is birds ,superman, messile, aeroplane use the same method that is fly. All of them reach the destination by flying .But have different speed and height and even food.