**ASSIGNMENT-3**

1. Given a class named **Patient,** identify what instance methods could be written in this class.

INSTANCE METHODS:

class Patient {

public void CreatePatientProfile();

public void PatientBill();

public void Patient In or Out();

public void CheckingPatient();

}

These are the instance methods so we can use these methods in main or another class

we need to create the object for patient and we call that methods

class main {

Patient p=new Patient();

p.CreatePatientProfile();

p.PayingPatientBill();

pPatientInOrOut ();

p.CheckingPatient ();

}

1. Given a class named **CPU,** identify the public & private instance members

CPU:

Public Access Modifier:

It can be specified by using the public keyword. Its scope or accessibility is the widest among other access specifiers.

The variables, classes, and methods declared as public can be accessed from everywhere in the program.

It does not impose restrictions on the scope of public data members. Coming to CPU point of view there is some Public access modifiers that are pendrive, cd, cables, power On/Off this actions or things can be access by any members.

public String pendrive;

public String powersupply;

public String cables;

public CD;

Private Access Modifier:

It is the opposite of the public modifier. It can be specified by using the private keyword followed by class name(applied only on nested classes) or method name or data member.

We cannot use the private access specifier with the top-level classes or interfaces.

Coming to CPU point of view there is some Private access modifiers like Clock speed, Logical units, cache these are only modified by particular member or developer only.

private logical units;

private control units;

private cache;

1. Given a class named **Media,** assume there are 3 objects of this class. Identify the static members of this class.

MEDIA:

In media there are some static members like photos, videos, documents etc. here we

created one class like media in this class we specify these static members

class Media {

public void static photos();

public void static videos();

public void static documents();

}

In main class we created three different objects like A,B,C by using these three objects we can access the static members.

class main {

Media A=new Media();

Media B=new Media();

Media C=new Media();

A.photos();

B.videos();

C.documents();

}

1. Given a class named **Calculator** & its derived classes named **Standard** and **Scientific,** identify the method which can be overridden by the derived classes.

CALCULATOR:

In both standard and scientific calculator there will be some common operation are there.

That is addition, subtraction, multiplication, division. So here we created the one parent class that is calculator with this common operations.

class Calculator{

void addition();

void subtraction();

void multiplication();

void division();

}

But there is some difference between standard calculator and scientific calculator.

Designed to help you calculate science, engineering and mathematics problems.

Here we created the one child class with name i.e., scientific calculator here we overridden the normal calculator methods which come to scientific calculator only

class scientific calculator extends calculator {

void addition();

void subtraction();

void multiplication();

void divition();

void engineering problems();

void scientific problems();

}

1. Given the classes **Bird, Superman, Aeroplane** and **Missile,** identify the usage of interfaces here.

Bird, Superman, Aeroplane, Missile:

The Common behaviour for all objects has “FLY”.

We created the one interface name called as Fly. In this Class the methods tells about that object behaviour.

Interface Fly {

void Fly()

}

Then comes to object the object states that there are two different states two objects are living things i.e., both bird and superman. So we created one class Living and it implements the Fly and in that we write two methods that are bird and superman

class Living implements Fly {

public void bird();

public void superman();

}

Similarly the Aeroplane and Missile are non living things so we created one class like non-living and it implements with fly, after that we wrote two methods like Aeroplane and missile

class Non\_Living implements Fly {

public void areoplane(0;

public void missile();

}

.