Apex using OOPs feature which is based on Object-oriented programming Language.

Object-oriented programming:is a software development programming approach that can be used to develop different types of applications like console based(CUI) ,windows based(GUI),web based and cloud based programming or mobile based programming**.**

Object-Oriented Programming:

Is a Software development approach using which we can develop different types of applications like console based(CUI),windows based(GUI), web based ,Cloud Based, Mobile based etc.

Features:

i)**Class :** is a collection of objects that share common attributes and behaviour.

Class is a blueprint/template that provides structure to the objects.

Example : Bird,Animal, Vehicle, Student etc

ii)**Object :** is an instance of a class.

Example: Peacock,Lion, Car, Ankit etc

Class-Vehicle

Attribute/variables: Color,NoOfTyres,Type

Behaviour/method/function: Accelerate,turn left,turn right,decelerate

Car

Auto

Truck

Bus

Attributes are represented using Variables and Behaviour of a class is represented using methods.

Class-Student

Attributes/variables: Name,Age,Address,Score,RollNo

Behaviour/Methods: DoRead(),DoStudy(),Assessment() etc

Object:

```
Student s1=new Student();
s1.Name='Ankit';
s1.Age=23;
s1.Address='Noida';
s1.Score=82;
s1.RollNo=1;

S1.DoRead();
s1.DoStudy();
s1.Assessment();

Student s2=new Student();
```

S2.Name='Aqib';

s2.Age=24;

s2.Address='New Delhi';

iii)Abstraction: Is a very important feature of OOP. Abstraction means showing only the necessary/important details/info.

Abstraction is implemented in programming using Abstract classes and Interface.

iv)**Encapsulation**: Is a very important feature of OOP. Encapsulation means hiding the unnecessary details.

Encapsulation is achieved using classes and properties.

Math.Mod(8,2);

v)Inheritance: Is a very important feature of OOP.

Inheritance means extending a class from another class.

This is done for reusability.

Parent Class/Base class/Super Class

Child Class/Derived Class/Sub Class

Two Seater Car=>Four Seater Car

i)Build a new Car from scratch

ii)Modify 2 seater and convert it into 4 seater.

Types:

i)Single Inheritance

ii)Multi-level inheritance

iii)Multiple Inheritance(Interface only)

iv)Hierarchical Inheritance

v)Hybrid Inheritance

vi)Polymorphism: Is a very important feature of OOP.

Polymorphism is a combination of Poly+ morphos which means many forms. Polymorphism is the ability of a thing to take many forms.

Polymorphism is achieved using Constructor Overloading,Operator Overloading,Function Overloading, Function Overriding.

Types:

i)**Static Polymorphism/Compile time Polymorphism/Early Binding:**

Constructor Overloading,Operator overloading, Function Overloading

Function Overloading works on

⇒No of arguments should be different

⇒No of type of arguments

ii)**Dynamic Polymorphism/Run time Polymorphism/Late Binding:**

Function Overriding(Virtual Class)

Access Modifiers: are used to define the scope/accessibility of a class , method or variables or data members.

Types:

i) **Private:** private members are accessible only inside a class. This is the default modifier for variables and methods.

ii) **Protected:** protected members are accessible inside a class in which they are defined or to the child classes.

iii) **Public:** Public members are accessible in every class within the salesforce.

iv) **Global:** Global members are accessible within salesforce org and to the applications outside the salesforce. In integration, a global access modifier is used.

Static Members :

=>are the members that get memory once throughout the program at the compilation time.

=>static members get memory on priority.

=>'static ' keyword is used to declare static members.

=>No need for objects to access static members they are called by the class name with the dot operator.

Classname.Membername;

=>static functions can only use static variables and static methods.

=>non-static functions can use both static and non-static variables.

=>Single copy of static members is shared by all the objects of the class.

⇒static method is only Public method

Class: Student

Variables:

string Name;-10 bytes

Integer RollNo;4 bytes

Function:void Study()-10 bytes

Student s1=new Student();-24 bytes

Student s2=new Student();-24 bytes

Constants :

=>are members of a class whose value is fixed and cannot be changed.

=> to make a constant member we need to use the **‘final’** keyword.

=> Constant members are initialised at the declaration time or inside a constructor only.

Ex: Pi=3.14

Final decimal Pi=3.14;

Final decimal GST=0.18;

Final integer x=900;

x=x+1; // error ,u r changing value of x is having a constant value

y=x+1; ///correct

Task[submit on 23]

⇒Questions to be done related to oops

⇒page no 14 read out till 28 oops concepts [dev].

⇒Fill in the blanks plus True and false assignments tomorrow also.

Example of Constructor Overloading

Example of Operator Overloading

Example of Property class

Constructor:A constructor is like an instance method that usually has the **same name as a class** and can be set to use the values of members of an object.

It is not any method since it does not have any return type.

⇒whenever an object is created an constructor is automatically called.

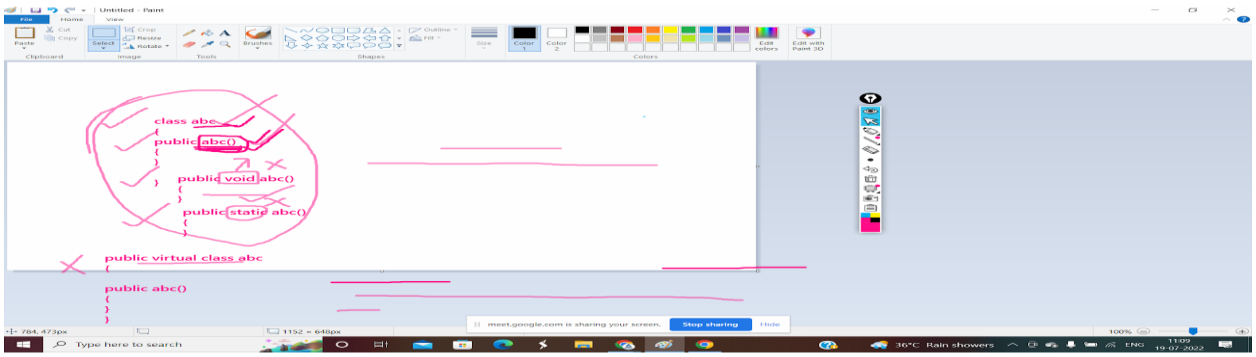
⇒users do not write constructors for every class.

⇒constructor cannot be inherited

⇒ **constructors always return some value so no void keyword is allowed.**

⇒constructor does not use any return data type.

⇒static is also not allowed in constructors.



```
class abc
{
public abc()
{
}
    public void abc()
    {
    }
}
    public static abc()
    {
    }

public virtual class abc
{
public abc()
{
}
}
```

Public class cclass// class name

{

public cclass// method name same as your class

{

}

public void cclass(integer x)// method name same as your class

{

}

}

Execution:

Cclass obj1=new cclass(30);

Cclass obj1=new cclass();

obj1.add();

Constants:

==>are the variables whose value is fixed and cannot be changed.

==>Final keyword is used to declare a constant variable.

Example

Final decimal p1=3.4;

Property class:

⇒is a class using get and set methods to take input from the user and to show a value.

⇒Property class is a functionality required when a user creates its own Interface.

⇒Propertyclass is use in VFpages