OOPS CONCEPT

Oops stands for : object oriented programming language

What is oops: oops is a programming language based on the concept of “objects”.the object contains both data and code data in the form of properties (often known as attributes),and code,in the form of methods (actions objects can perform.

OBJECTS

ENCAPSULATION ABSTRACTION

OOPS CONCEPT

POLYMORPHISM INHERITANCE

CLASS

Oops terminology:

attributes also known as data member or variables

Behavior also known as member function or method

What is class: class is defined with keywords class.the class is a user defined data structures that binds the data members and mrthods into a single unit.class is a biue print or code template for object creation . using a class ,you can create as many objects s you want.

example: Mercedes,BMW,Toyota,etc.

class syntax= class classname():

\*\*class name first letter should be capital\*\*

example: class Prami():

print(“this is class”)

what is object?

an objectis simply a collection of data (variables)and methods(functions) that act on those data .similarly,a class is a blueprint for the object.

the object is the instance of a class .the process of creating an object can be called instantiation.

there is no memory allocation until we create its object.the object instance contains real data or information.

example: chair,bike,marker,pen,table,car etc.

# object syntax: object name = class name()

ex: class kiran:

a=11

def output(self):

print(“this is class”)

bhavan=kiran()

#object(variable+function)

#obj name.variable

#obj.method

bhavan.output()

print(bhavan.a)

self keyword:

the self parameter is used to access current class methods and variables. It does not have to be named self, you can call it whatever you like, but it has to be the first parameter of any function I the class.

# what is \_\_init\_\_?

\_\_init\_\_ method in oop is nothing but a special function.special functions are the functions that are used to enrich the class.these can be easily identified as they have double underscore on either side: \_\_init\_\_mrthod is used to initialize the attributes .it is called a constructor in other programming languages.

INHERITANCE:

inheritance allows us to define a class that inherits all the methodsand properties from another class.

parent class is the class being inherited from , also called base class.

child class is the class that inherits from another class, also called derived class.

types of inheritance :

1. single\_ inheritance
2. multilevel\_ inheritance
3. hierarchial\_ inheritance
4. multiple\_inheritance

CLASS A

class A

CLASS B

class B

CLASS C

SINGLE INHERITANCE

MULTILEVEL INHERITANCE

CLASS A

CLASS B

CLASS A

CLASS C

CLASS C

CLASS B

HIERARCHICAL INHERITANCE MULTIPLE INHERITANCE

POLYMORPHISM: The literal meaning of polymorphism is the condition of occurrence in different forms:

there are two types:

# METHOD OVERLOADING

# METHOD OVER RIDING

TO OVERCOME METHOD OVERLOADING WE HAVE TO USE “None”

TO OVERCOME METHOD OVER RIDING WE HAVE TO USE “super().display()”

ENCAPSULATION:

Encapsulation is a mechanism of wrapping the data (variable) and code acting on the data (methods) together as a single unit.

ACCESS MODIFIERS:

1. PUBLIC
2. \_PROTECTED (SINGLE UNDERSCORE)
3. \_\_PRIVATE (DOUBLE UNDERSCORE)

DATA ABSTRACTION:

These methods are unimplemented in the parent class

\*\* only definition will be written in the parent class.

in the child classes, we can provide the implementations.

ex: def taste(): def taste ():

pass print(“tangy”)

def color(): def color():

pass print(“tangerine”)

## parent class. ## child class

def taste ():

print(“sweet”)

def color():

print(“yellow”)

## child class

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*