**Practical No: 21**

**Aim:** Classes and Objects: Write a Program to: Create a Class which Performs Basic Calculator Operations

**Course Outcome:** Write ‘Python’ code using Classes and Objects.

**Requirements: Computer, Python 3.3.34, Vs Code.**

**Theory:**

A class is a user-defined blueprint or prototype from which objects are created. Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by their class) for modifying their state.

To understand the need for creating a class let’s consider an example, let’s say you wanted to track the number of dogs that may have different attributes like breed, age. If a list is used, the first element could be the dog’s breed while the second element could represent its age. Let’s suppose there are 100 different dogs, then how would you know which element is supposed to be which? What if you wanted to add other properties to these dogs? This lacks organization and it’s the exact need for classes.

Class creates a user-defined data structure, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A class is like a blueprint for an object.

The self

Class methods must have an extra first parameter in the method definition. We do not give a value for this parameter when we call the method, Python provides it.

If we have a method that takes no arguments, then we still have to have one argument.

This is similar to this pointer in C++ and this reference in Java.

When we call a method of this object as myobject.method(arg1, arg2), this is automatically converted by Python into MyClass.method(myobject, arg1, arg2) – this is all the special self is about.

\_\_init\_\_ method

The \_\_init\_\_ method is similar to constructors in C++ and Java. Constructors are used to initializing the object’s state. Like methods, a constructor also contains a collection of statements(i.e. instructions) that are executed at the time of Object creation. It runs as soon as an object of a class is instantiated. The method is useful to do any initialization you want to do with your object.

**Flowchart:**

**Program:**

class Calculator():

    def \_\_init\_\_(self, x,y):

        self.First = x

        self.Second = y

    def add(self):

        print(self.First + self.Second)

    def sub(self):

        print(self.First - self.Second)

    def mul(self):

        print(self.First \* self.Second)

    def div(self):

        print(self.First / self.Second)

First = int(input("Enter First Number: "))

Second = int(input("Enter Second Number: "))

Obj = Calculator(First,Second)

los = 0

while True:

    menu = ('1. Add \n2 .Subtract \n3. Multiply \n4. Divide ')

    print(menu)

    i = int(input('Enter your choice: '))

    if i == 1:

        Obj.add()

    elif i == 2:

        Obj.sub()

    elif i == 3:

        Obj.mul()

    elif i == 4:

        Obj.div()

    elif i == 0:

        print('Thank You....')

        break

    else:

        print('Enter appropriate option...')

**Output/Result:**

Enter First Number: 10

Enter Second Number: 20

1. Add

2 .Subtract

3. Multiply

4. Divide

Enter your choice: 1

30

1. Add

2 .Subtract

3. Multiply

4. Divide

Enter your choice: 02

-10

1. Add

2 .Subtract

3. Multiply

4. Divide

Enter your choice: 0

Thank You....

**Conclusion:** In this practical we have first created a class named Calculator. For passing the values to the functions we have used the \_\_init\_\_() constructor and Self parameter to get the objects result. Then created different which performs arithmetic operations such as add , sub etc.. . this function will give us the result as the output. Then initiated an object which takes the user input as parameters. For looping our options repeatedly, we have used the while loop which checks the condition and give the output as per the options. Hence, we have created a basic calculator using class and objects.