**Practical No: 21**

**Aim:** Classes and Objects: Write a Program to: Add two complex number using classes and objects. Subtract two complex number using classes and objects

**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 Complex ():

    def initComplex(self):

        self.realPart = int(input("Enter the Real Part: "))

        self.imgPart = int(input("Enter the Imaginary Part: "))

    def display(self):

        print(self.realPart,"+",self.imgPart,"i", sep="")

    def sum(self, c1, c2):

        self.realPart = c1.realPart + c2.realPart

        self.imgPart = c1.imgPart + c2.imgPart

    def Sub(self, c1, c2):

        self.realPart = c1.realPart - c2.realPart

        self.imgPart = c1.imgPart - c2.imgPart

c1 = Complex()

c2 = Complex()

c3 = Complex()

print("Enter first complex number")

c1.initComplex()

print("First Complex Number: ", end="")

c1.display()

print("Enter second complex number")

c2.initComplex()

print("Second Complex Number: ", end="")

c2.display()

while True:

    menu = '1. Add Complex Number... \n2. Subtract Complex Number... '

    print(menu)

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

    if i == 1:

        print("Sum of two complex numbers is ", end="")

        c3.sum(c1,c2)

        c3.display()

    elif i == 2:

        print("Subtraction of two complex numbers is ", end="")

        c3.Sub(c1,c2)

        c3.display()

    elif i == 0:

        break

    else:

        print('Thank You...')

**Output/Result:**

Enter first complex number

Enter the Real Part: 10

Enter the Imaginary Part: 4

First Complex Number: 10+4i

Enter second complex number

Enter the Real Part: 5

Enter the Imaginary Part: 2

Second Complex Number: 5+2i

1. Add Complex Number...

2. Subtract Complex Number...

Enter your choice: 2

Subtraction of two complex numbers is 5+2i

1. Add Complex Number...

2. Subtract Complex Number...

Enter your choice: 1

Sum of two complex numbers is 15+6i

1. Add Complex Number...

2. Subtract Complex Number...

Enter your choice: 0

Thank You....

**Conclusion:** In this practical we have created a class named Complex. In that we have initiated the init constructor to get the real and imaginary part of the complex number. Then created a display method which will display the complex number. Then created two functions add and sub to add and subtract the complex number. In these functions we first add or subtract the real part and imaginary part separately. To display full complex number, we used the display method. Then initiated three objects c1, c2, c3 the c1 will hold the first complex number and c2 will hold Second complex number. C3 will be used to store the result. Then we have used while loop for repeating option for addition and subtraction. Hence, we have added and subtracted complex numbers using Classes and objects.