Computer Programming Laboratory

B.Tech. 1st Semester

Name   : SUBHENDU MAJI

Roll Number   : 18ETCS002121

Department   : Computer Science and Engineering

Faculty of Engineering & Technology
Ramaiah University of Applied Sciences

Faculty	Engineering & Technology
Programme	B. Tech. in Computer Science and Engineering
Year/Semester	1st Year / 1st Semester
Name of the Laboratory	Computer Programming Laboratory
Laboratory Code	18ESL109A

List of Experiments

1. Introduction to Python programming environment
2. Variables, data types, operators and expressions
3. Input output operations
4. Logic operations and decision making
5. Loop statements
6. Character and string operations
7. Functions
8. File handling
9. Data structures
10. Libraries

Index Sheet

No.	Lab Experiment	Performing the experiment (7)	Document (7)	Viva (6)	Total Marks (20)
1	Introduction to Python programming environment				
2	Variables, data types, operators and expressions				
3	Input output operations				
4	Logic operations and decision making				
5	Loop statements				
6	Character and string operations				
7	Functions				
8	File handling				
9	Data structures				
10	Libraries				
11	Lab Internal Test conducted along the lines of SEE and valued for 50 Marks and reduced for 20 Marks				
	Total Marks				

Lab Internal Marks =

Signature of the Staff In-charge

Laboratory 3

Title of the Laboratory Exercise: Input Output operations

1. Introduction and Purpose of Experiment

   Python provides the Input-Output facility to interact with the user. Some of the functions such as input() and print() are widely used for standard input and output operations respectively. By solving the problems, students will be able to develop programs to read input from screen and display output to the screen.

2. Aim and Objectives

   Aim

   - To develop programs involving loops and branching using appropriate Python language control statements

   Objectives

   At the end of this lab, the student will be able to

   - Apply control statements such as if-else, nested if-else to express decisions
   - Use the switch statement to create multiple branching based on expression matching
   - Create C programs using loops such as for, while, do-while to repeat a block of code

3. Experimental Procedure

   i. Analyse the problem statement

   ii. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code

   iii. Implement the algorithm in Python language

   iv. Execute the Python program

   v. Test the implemented program

   vi. Document the Results

   vii. Analyse and discuss the outcomes of the experiment

4. Questions

   a. Write a program to display the square root of a given integer number

      i. Using built in method

      ii. Without using built in method

b. Write a program to accept two complex numbers from the user and display their sum, difference and product

c. Write a program to accept three numbers from the user (in a single line) and display their average

5. Algorithms

5.1 Algorithm of a program to display the square root of a given integer number

5.1.1   Using built in method

step1: start

step2: read a variable number of type int

step3: sqr := math.sqrt(number)

step4: write the value of sqr

step5: stop

5.1.2 Without using built in method

step1: start

step2: read a variable number of type int

step3: sqr := number**0.5

step4: write the value of sqr

step5: stop

5.2 Algorithm of a program to accept two complex numbers from the user and display their sum, difference and product

step1: start

step2: read four variables a1, b1, a2, b2 of type int

step3: complex1 := complex(a1,b1)

complex2 := complex(a2,b2)

step4:   add := complex1 + complex2

dif := complex1 - complex2

prod := complex1 * complex2

step5: write the value of sum, dif and prod

step6: stop

5.3 Algorithm of a program to accept three numbers from the user (in a single line) and display their average

        step1: start

        step2: read three variables a, b, c of type int

        step3: x := int(a)

           y := int(b)

           z := int(c)
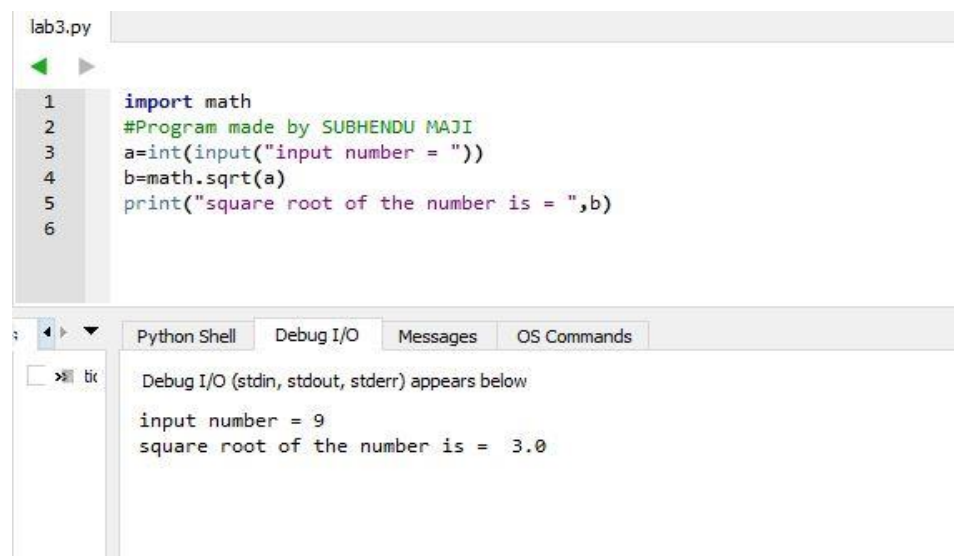
        step4: total := x+y+z

        step5: avg := total/3

        step6: write the value of total and avg

        step7: stop

6. Presentation of Results
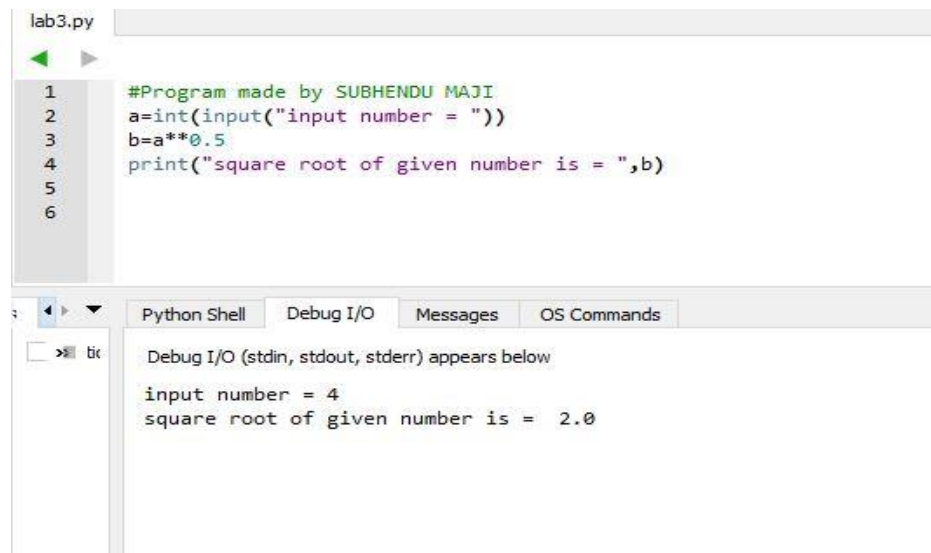
```
lab3.py

1    import math
2    #Program made by SUBHENDU MAJI
3    a=int(input("input number = "))
4    b=math.sqrt(a)
5    print("square root of the number is = ",b)
6

Python Shell    Debug I/O    Messages    OS Commands

Debug I/O (stdin, stdout, stderr) appears below

input number = 9
square root of the number is =  3.0
```

Figure 6.1 Output of program to display the square root of a given integer number
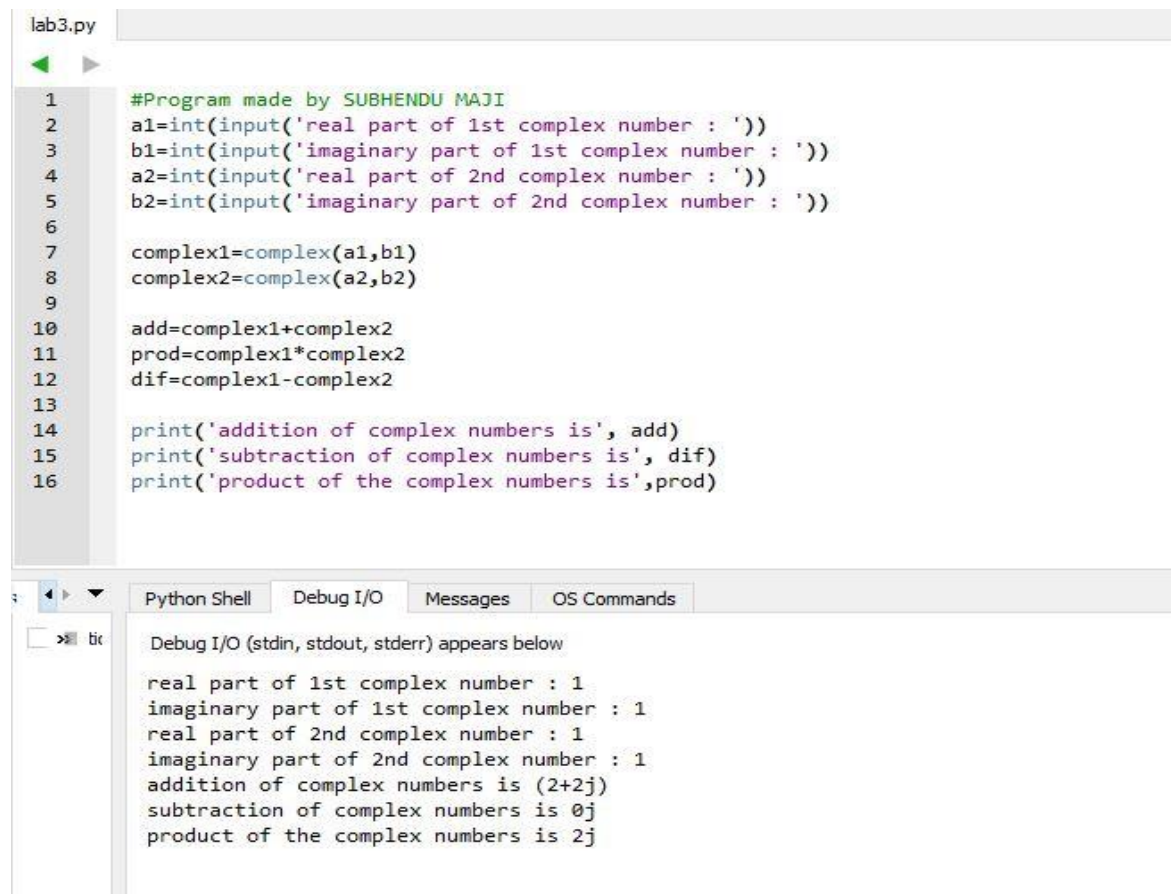
using built in method

lab3.py

◄ ►

```python
1    #Program made by SUBHENDU MAJI
2    a=int(input("input number = "))
3    b=a**0.5
4    print("square root of given number is = ",b)
5
6
```

| Python Shell | Debug I/O | Messages | OS Commands |

☐ >≡ tic   Debug I/O (stdin, stdout, stderr) appears below

```
input number = 4
square root of given number is =  2.0
```

Figure 6.2 Output of program to display the square root of a given integer number
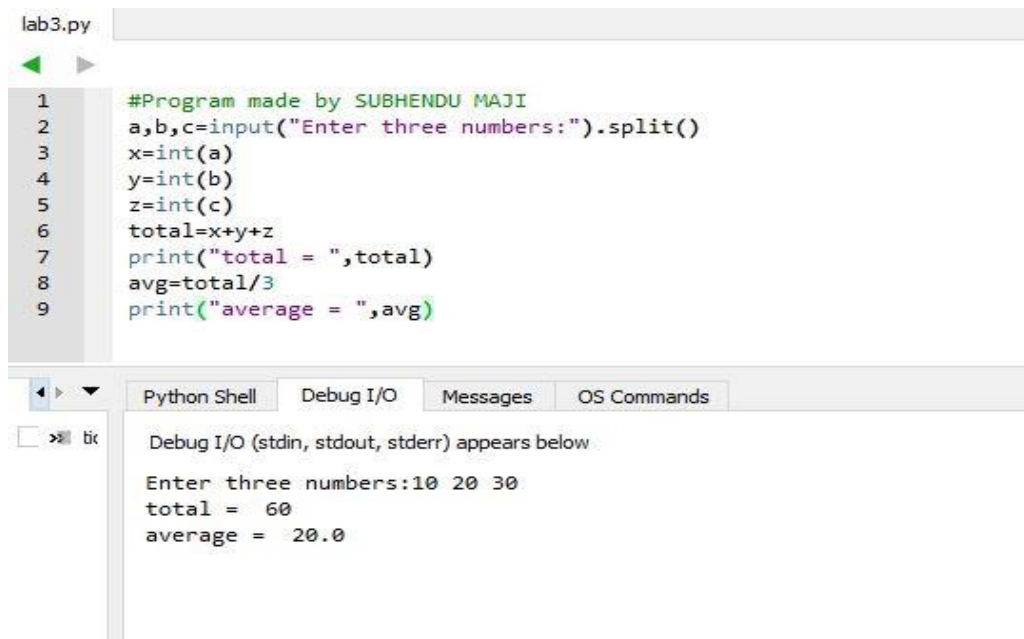
without using built in method

lab3.py

◄ ►

```python
1    #Program made by SUBHENDU MAJI
2    a1=int(input('real part of 1st complex number : '))
3    b1=int(input('imaginary part of 1st complex number : '))
4    a2=int(input('real part of 2nd complex number : '))
5    b2=int(input('imaginary part of 2nd complex number : '))
6
7    complex1=complex(a1,b1)
8    complex2=complex(a2,b2)
9
10   add=complex1+complex2
11   prod=complex1*complex2
12   dif=complex1-complex2
13
14   print('addition of complex numbers is', add)
15   print('subtraction of complex numbers is', dif)
16   print('product of the complex numbers is',prod)
```

| Python Shell | Debug I/O | Messages | OS Commands |

☐ >≡ tic   Debug I/O (stdin, stdout, stderr) appears below

```
real part of 1st complex number : 1
imaginary part of 1st complex number : 1
real part of 2nd complex number : 1
imaginary part of 2nd complex number : 1
addition of complex numbers is (2+2j)
subtraction of complex numbers is 0j
product of the complex numbers is 2j
```

Figure 6.3 Output of program to accept two complex numbers from the user and

display their sum, difference and product

```
lab3.py

1    #Program made by SUBHENDU MAJI
2    a,b,c=input("Enter three numbers:").split()
3    x=int(a)
4    y=int(b)
5    z=int(c)
6    total=x+y+z
7    print("total = ",total)
8    avg=total/3
9    print("average = ",avg)
```

```
Python Shell    Debug I/O    Messages    OS Commands

Debug I/O (stdin, stdout, stderr) appears below

Enter three numbers:10 20 30
total =   60
average =   20.0
```

Figure 6.4 Output of program to accept three numbers from the user (in a single line) and display their average

7. Analysis and Discussions

7.1 Algorithm of a program to display the square root of a given integer number

7.1.1   Using built in method

First, user inputs the number of which he wants to calculate the square root. The program uses the built-in function 'sqrt' to calculate the square root of the given number. Then the result is printed.

7.1.2   Without Using built in method

First, user inputs the number of which he wants to calculate the square root. The program manually calculates the square root of the given number using the formula

$$square\ root = number ** 0.5$$

Then the result is printed.

7.2 Algorithm of program to accept two complex numbers from the user and display their sum, difference and product

The program takes inputs of the real part and imaginary part of the first number, then for the second number. Both the numbers are type casted to complex number. Now, using the formula

$$sum = complex1 + complex2$$
$$product = complex1 * complex2$$
$$differernce = complex1 - complex2$$

the sum, difference and product is calculated. Then the result is printed.

7.3 Algorithm of program to accept three numbers from the user (in a single line) and display their average

For the user to input three variables separated by space, using of split() function is mandatory. Then, we have to type cast the variables to int type. The average of the numbers is calculated by the following formula

$$average = \frac{a + b + c}{3}$$

Hence, the result is printed.

7   Conclusions

Programs involving loops and branching using appropriate Python language control statements is practiced.

8   Comments

1. Limitations of Results

The program 5.1 takes only integer value for calculating square root. If the user inputs decimal value, interpreter will show value error. The program 5.3 also takes only integer value for calculating average.