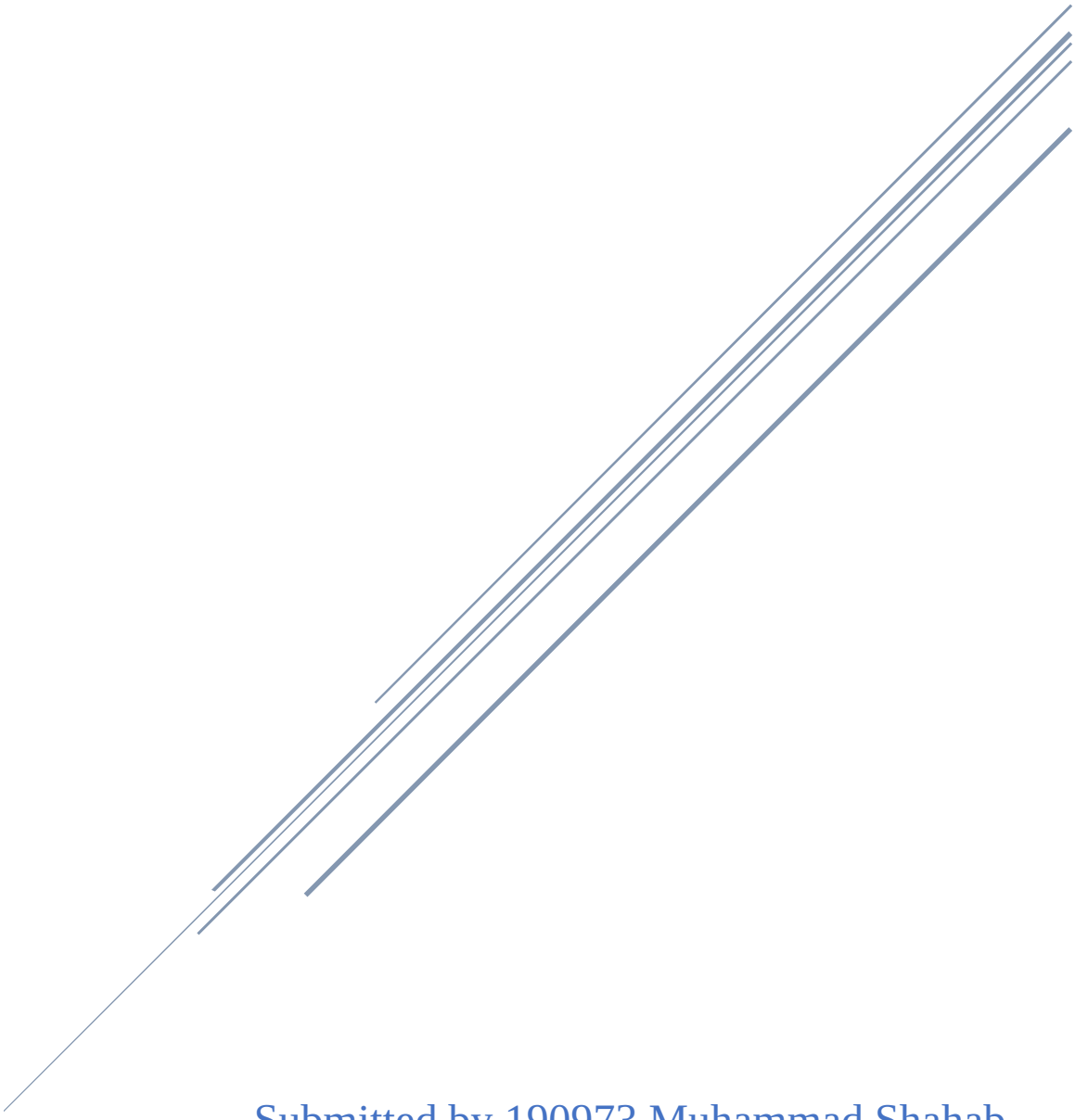
MACHINE LEARNING LAB

Department of Mechatronics Engineering



Submitted by 190973 Muhammad Shahab
Submitted to Ma’am Iraj Kainaat

Lab#1: Introduction to Python Fundamentals

Objective:

- Familiarize students with the Python language.
- To teach students the conventional coding practices in Python.

Introduction:

Python is a high-level general purpose programming language. Because code is automaticallycompiled to byte code and executed, Python is suitable for use as a scripting language, Web application implementation language, etc. Because Python can be extended in C and C++, Python can provide the speed needed for even computing intensive tasks.

- Contains in-built sophisticated data structures like strings, lists, dictionaries, etc.
- The usual control structures: if, if-else, **if-elif-else**, while, plus a powerful collection iterator (for).
- Multiple levels of organizational structure: functions, classes, modules, and packages. They assist in organizing code. An excellent and large example is the Python standard library.
- Python uses indentation to show block structure. Indent one level to show the beginningof a block.

Programming Environment:

An IDE (or Integrated Development Environment) is a program dedicated to software development. As the name implies, IDEs integrate several tools specifically designed for software development. These tools usually include:

- An editor designed to handle code (with, for example, syntax highlighting and auto-completion)
- Build, execution, and debugging tools.
- Some form of source control

Most IDEs support many different programming languages and contain many more features. They can, therefore, be large and take time to download and install. You may also need advanced knowledge to use them properly.

In contrast, a dedicated code editor can be as simple as a text editor with syntax highlighting and code formatting capabilities. Most good code editors can execute code and control a debugger. The very best ones interact with source control systems as well. Compared to an IDE, a good, dedicated code editor is usually smaller and quicker, but often less feature rich.

Lab Tasks:

Question #1:

Write a simple calculator program**.**

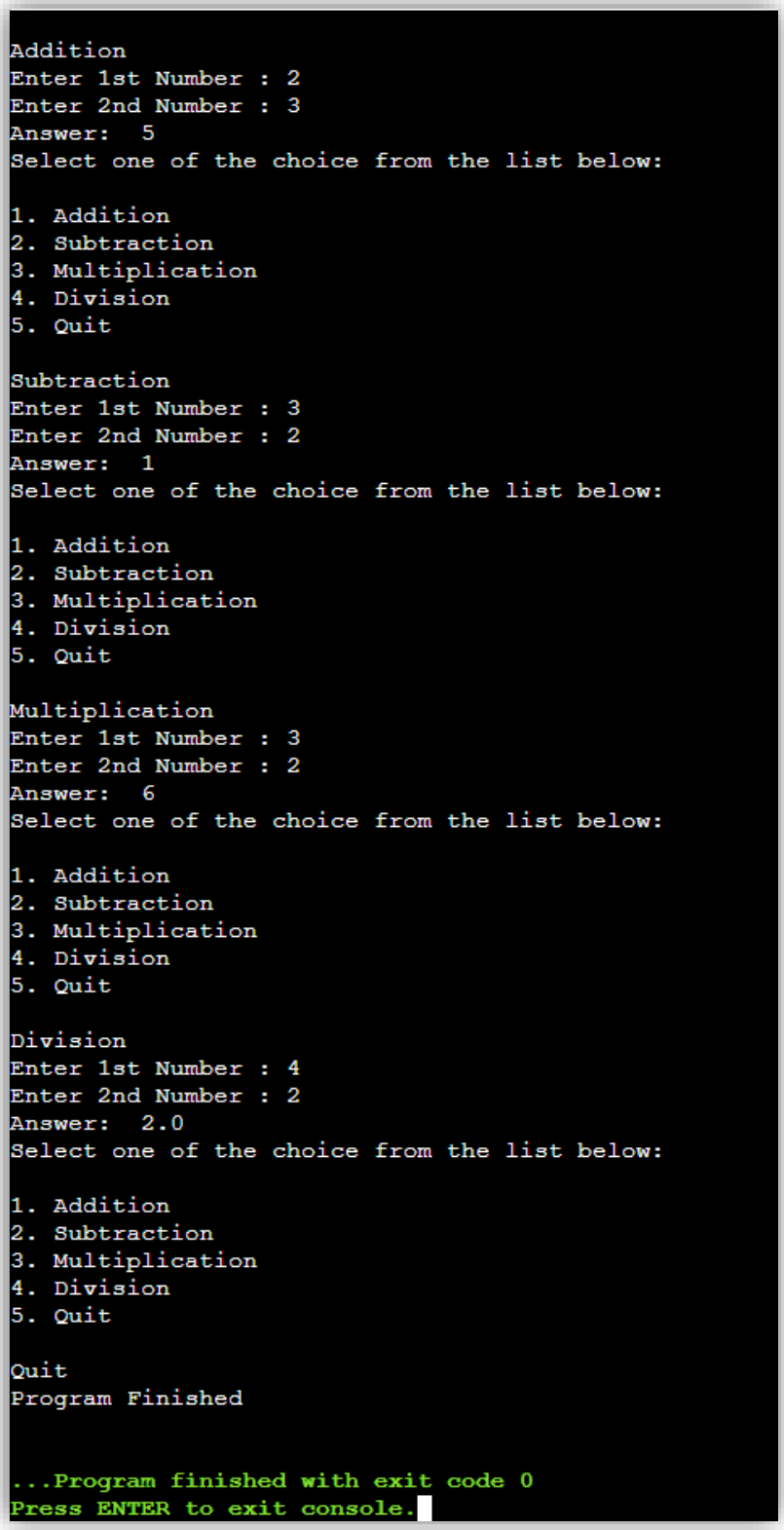
Code Snippet:

```
def addition():
    a = int(input("Enter 1st Number : "))
    b = int(input("Enter 2nd Number : "))
    print()
    return a+b #return the summation of two number
def subtraction():
    a = int(input("Enter 1st Number : "))
    b = int(input("Enter 2nd Number : "))
    return a-b #return after subtracting number
def multiplication():
    a = int(input("Enter 1st Number : "))
    b = int(input("Enter 2nd Number : "))
    return a*b #returns the product
def division():
    a = int(input("Enter 1st Number : "))
    b = int(input("Enter 2nd Number : "))
    return a/b #return the quotient



inp = "Initial" #value initialized
while((inp != "Quit") and (inp != "quit")):#continues till Quit
    print("Select one of the choice from the list below:\n")
    print("1. Addition\n2. Subtraction\n3. Multiplication\n4. Division\n5. Quit\n")
    inp = str(input())
    if(inp == "Addition"):
        print("Answer: ", addition())
    elif(inp == "Subtraction"):
        print("Answer: ", subtraction())
    elif(inp == "Multiplication"):
        print("Answer: ", multiplication())
    elif(inp == "Division"):
        print("Answer: ", division())
    elif(inp == "Quit"):
        print("Program Finished")
    else:
        print("Not Valid Input.ReType")#exception handled
```

Output:


Addition
Enter 1st Number : 2
Enter 2nd Number : 3
Answer: 5
Select one of the choice from the list below:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Quit
Subtraction
Enter 1st Number : 3
Enter 2nd Number : 2
Answer: 1
Select one of the choice from the list below:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Quit
Multiplication
Enter 1st Number : 3
Enter 2nd Number : 2
Answer: 6
Select one of the choice from the list below:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Quit
Division
Enter 1st Number : 4
Enter 2nd Number : 2
Answer: 2.0
Select one of the choice from the list below:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Quit
Quit
Program Finished
...Program finished with exit code 0
Press ENTER to exit console.

Question #2:
Find the factorial of the given number.

Code Snippet:

```python
#factorial using Recursion
def factorial(num):
    if(num <=1): #base case
        return 1
    else:
        return num*factorial(num-1) #recursion

inp = int(input("Enter number to find factorial : "))
print(factorial(inp))
```

Output:

```
Enter number to find factorial : 4
24


...Program finished with exit code 0
Press ENTER to exit console.
```

Conclusion:

In this lab, we have reviewed the syntax of python programming language. The concepts are similar to what we have learned in C++. Just the addition of tuples and Dictionaries was new and interesting. By the end of this lab we were able to write optimized program for the given tasks.