LAB # 01

Statement Purpose:

This lab is an introductory session on **Python**. It is a powerful object-oriented programming language, comparable to Perl, Ruby, Scheme and Java. Some of Python's notable features are:

- Easy to use that makes it simple to get your first programs working
- Easy to learn that makes it an excellent choice for beginners
- Runs everywhere, including Mac OS X, Windows, Linux and Unix

Activity Outcomes:

The lab will teach students to:

- Basic Operations on strings and numbers
- Basic use of conditionals
- Basic use of loops

Instructor Note:

As a pre-lab activity, read Chapters 3-5 from the book (*Introduction to Programming Using Python - Y. Liang (Pearson, 2013)*) to gain an insight about python programming and its fundamentals.

1) Stage J(Journey)

Introduction

Programming is simply the act of entering instructions for the computer to perform. These instructions might crunch some numbers, modify text, look up information in files, or communicate with other computers over the Internet. All programs use basic instructions as building blocks. Here are a few of the most common ones, in English:

- "Do this; then do that."
- "If this condition is true, perform this action; otherwise, do that action."
- "Do this action that number of times."
- "Keep doing that until this condition is true."

You can combine these building blocks to implement more intricate decisions, too. Programming is a creative task, somewhat like constructing a castle out of LEGO bricks. You start with a basic idea of what you want your castle to look like and inventory your available blocks. Then you start building. Once you've finished building your program, you can pretty up your code just like you would your castle.

Python refers to the Python programming language (with syntax rules for writing what is considered valid Python code) and the Python interpreter software that reads source code (written in the Python language) and performs its instructions. The name Python comes from the surreal British comedy group Monty Python, not from the snake. Python programmers are affectionately called Pythonistas.

2) Stage a1 (apply)

Lab Activities:

Activity 1:

Use Python as a calculator.

Solution:

1. The interpreter acts as a simple calculator: you can type an expression at it and it will write the value. Expression syntax is straightforward: the operators + (addition), - (subtraction), * (multiplication) and / (division) work just like in most other languages. Parentheses (()) can be used for grouping.
2. Type in **2 + 2** and press **Enter**. Python will output its result i.e. **4**.
3. Try following expressions in the shell.
    a. 50 – 4

b. 23.5 – 2.0

c. 23 – 18.5

d. 5 * 6

e. 2.5 * 10

f. 2.5 * 2.5

g. 28 / 4 (note: division returns a floating point number)

h. 26 / 4

i. 23.4 / 3.1

Activity 2:

Combine numbers and text.

Solution:

Type the following code. Run it.

```
# Text
x = "Nancy"
print(x)

# Combine numbers and text
s = "My lucky number is %d, what is yours?" % 7
print(s)

# alternative method of combining numbers and text
s = "My lucky number is " + str(7) + ", what is yours?"
print(s)
```

Activity 3:

Let us take an integer from user as input and check whether the given value is even or not.

Solution:

A. Create a new Python file from Python Shell and type the following code.

B. Run the code by pressing F5.

```
n=input("Enter a number ")
if int(n)%2==0:
    print("The given number is an even number")
```

You will get the following output.

```
Enter a number 12
The given number is an even number
>>>
```

Activity 4:

Let us modify the code to take an integer from user as input and check whether the given value is even or odd. If the given value is not even then it means that it will be odd. So here we need to use if-else statement an demonstrated below.

Solution:

A. Create a new Python file from Python Shell and type the following code.
B. Run the code by pressing F5.

```
n=input("Enter a number ")
if int(n)%2==0:
    print("The given number is an even number")
else:
    print("The given number is an odd number")
```

You will get the following output.

```
Enter a number 11
The given number is an odd number
>>>
```

Activity 5:

Calculate the sum of all the values between 0-10 using while loop.

Solution:

- Create a new Python file from Python Shell and type the following code.
- Run the code by pressing F5.

```
summation = 0
i=1
while i<=10:
    summation=summation+i
    i=i+1

print("sum is ",summation)
```

You will get the following output.

```
sum is  55
>>>
```

Activity 6:

Write a Python code to keep accepting integer values from user until 0 is entered. Display sum of the given values.

Solution:

- Create a new Python file from Python Shell and type the following code.
- Run the code by pressing F5.

```
sum=0
s=input("Enter an integer value...")
n=int(s)
while n!=0:
    sum=sum+n
    s=input("Enter an integer value...")
    n=int(s)
print("Sum of given values is ",sum)
```

You will get the following output.

```
Enter an integer value...10
Enter an integer value...521
Enter an integer value...5
Enter an integer value...22
Enter an integer value...0
Sum of given values is  558
>>>
```

Activity 7:

Write a Python code to accept an integer value from user and check that whether the given value is prime number or not.

Solution:

- Create a new Python file from Python Shell and type the following code.
- Run the code by pressing F5.

```
isPrime = True
i=2
n=int(input("enter a number"))
while i<n:
    remainder=n%i
    if remainder==0:
        isPrime=False
        break
    else:
        i=i+1

if isPrime:
    print("Number is Prime")
else:
    print("Number is not Prime")
```

You will get the following output(s).

```
enter a number17
Number is Prime
>>>
```

```
enter a number49
Number is not Prime
>>>
```

3) Stage a2 (Asses)

Graded Activities:

Activity 1:

Write a Python code to accept marks of a student from 1-100 and display the grade according to the following formula.

Grade F if marks are less than 50 Grade E if marks are between 50 to 60 Grade D if marks are between 61 to 70 Grade C if marks are between 71 to 80 Grade B if marks are between 81 to 90
Grade A if marks are between 91 to 100

Activity 2:

Write a program that takes a number from user and calculate the factorial of that number.