LAB # 01

INTRODUCTION TO PYTHON , OPERATOR AND STRING

OBJECTIVE

Familiarization with Python language using operator and string.

THEORY

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is widely used for Artificial Intelligence, with packages for a number of applications including Machine Learning and NLP. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- **Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python Features:

Python's features include:

- **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.

- **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.

- **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

- **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable:** Python provides a better structure and support for large programs than shell scripting.

Modules in Python:

- A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a file consisting of Python code. A module can define functions, classes and variables.
- In Python, modules are accessed by using the 'import' statement. When you do this, you execute the code of the module, keeping the scopes of the definitions so that your current file(s) can make use of these.
- Many build-in modules of python, some of above

1. Math
2. Random
3. Fraction
4. Decimal
5. OS

Operator in Python:

- Basic algebraic operations
  - ✓ Four arithmetic operations: a+b, a-b, a*b, a/b
  - ✓ Exponentiation: a**b
  - ✓ Other elementary functions are not part of standard Python, but included in packages like NumPy and SciPy
- Comparison operators
  - ✓ Greater than, less than, etc.: a < b, a > b, a <= b, a >= b
  - ✓ Identity tests: a == b, a != b
- Bitwise operators
  - ✓ Bitwise or: a | b
  - ✓ Bitwise exclusive or: a ^ b # Don't confuse this with exponentiation
  - ✓ Bitwise and: a & b
  - ✓ Shift a left or right by b bits: a << b, a >> b

Input( ) Function:

The input() function pauses your program and waits for the user to enter some text. Once Python receives the user's input, it stores it in a variable to make it convenient for you to work with.
The purpose of an input statement is to get some information from the user of a program and store it into a Variable.

- Syntax: <variable> = input (<prompt>)

Example:

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AI
 on win32
Type "copyright", "credits" or "license()" for more information.
>>> message = input("Tell me something, and I will repeat it back to you: "
Tell me something, and I will repeat it back to you: Hello everyone!
>>> print(message)
Hello everyone!
>>>
```

Strings:

A *string* is simply a series of characters. Anything inside quotes is considered a string in Python, and you can use single or double quotes around your strings like this:
"This is a string."
'This is also a string.'

Changing Case in a String with Methods
One of the simplest tasks you can do with strings is change the case of the words in a string. Look at the following code, and try to determine what's happening:

```
>>> name = "string python"
>>> print(name.title())
String Python
>>>
```

Combining or Concatenating Strings:
When applied to strings, the + operation is called concatenation. It produces a new string that is a copy of the two original strings pasted together end-to-end. Notice that concatenation doesn't do anything clever like insert a space between the words. The Python interpreter has no way of knowing that you want a space; it does exactly what it is told.

```
>>> 'Hello'+'World'
'HelloWorld'
>>> 'Hi'+'Hi'+'Hi'
'HiHiHi'
>>> 'Hi'*3
'HiHiHi'
>>>
```

Strings can be concatenated with the '+' operator and repeated with '*'

Example:

```
>>> first_name = "sammy"
>>> last_name ="more"
>>> full_name = first_name + " " + last_name
>>> print("Hello, " + full_name.title() + "!")
Hello, Sammy More!
>>>
```

Indexing of string:

Python starts indexing at 0. A string s will have indexes running from 0 to len(s)-1 (where len(s) is the length of s) in integer quantities.

S	A	m	m	Y		S	h	A	r	k	!
0	1	2	3	4	5	6	7	8	9	10	11

Example:

- s[i] : fetches the ith element in s
- s[i:j] : fetches elements i (inclusive) through j (not inclusive)
- s[:j] fetches all elements up to, but not including j
- s[i:] fetches all elements from i onward (inclusive)
- s[i:j:k] extracts every kth element starting with index i (inlcusive) and ending with index j (not inclusive)
- Python also supports negative indexes. For example, s[-1] means extract the first element of s from the end (same as s[len(s)-1])

```
ss=”Sammy Shark! “
print(ss[2])
print(ss[6:11])
print(ss[:5])
print(ss[5:])
print(ss[0:5:2])
print(ss[-2])
```

Output:

```
m
Shark
Sammy
 Shark!
Smy
k
>>> 
```

Lab Task:

1. Write a script that take user input for a number then adds 3 to that number. Then multiplies the result by 2, subtract 4, then again adds 3, then print the result.

2. Write a script that takes input as Celsius and then convert Celsius to Fahrenheit. (hint: Fahrenheit = (Celsius * 1.8) + 32)

3. Write a script that takes input as radius then calculate area of circle. (hint: $A = \pi r^2$)

4. Write a Python script that asks users for their favourite color. Create the following output (assuming blue is the chosen color) (hint: use ‘+’ and ‘*’)

```
blueblueblueblueblueblueblueblueblueblue
blue                                blue
blueblueblueblueblueblueblueblueblueblue
```

5. Store a person’s name, and include some whitespace characters at the beginning and end of the name. Make sure you use each character combination, "\t" and "\n", at least once. Print the name once, so the whitespace around the name is displayed. Then print the name using each of the three stripping functions, lstrip( ), rstrip(), and strip().