Lecture 4

Variable Names

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume). Rules for Python variables:

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (age, Age and AGE are three different variables)

Assign Value to Multiple Variables

Python allows you to assign values to multiple variables in one line.

Eg: x, y, z = "Orange", "Banana", "Cherry"

print(x)

print(y)

print(z)

 # Orange

 # Banana

 # Cherry

```
# represents comment for a line and for a code block specify within ' ' ' and ' ' '
```

You can assign the same value to multiple variables in one line.

Eg:

x = y = z = "Orange"

print(x)

print(y)

print(z)

 # Orange

 # Orange

 # Orange

To combine text and a variable, Python uses the + character.

You can also use the + character to add a variable to another variable.

Eg1:

```
x = "Python is "
y = "awesome"
z =  x + y
print(z)      # Python is awesome
```

Eg2:

```
x = 5
y = 10
print(x + y)  # 15
```

Eg3:

```
x = 5
y = "John"
print(x + y)  # Error
```

Keywords: Eg: def, if, else, for, global etc…..

Two built-in functions print() and input() to perform I/O task in Python.

Python Indentation: Programming languages like C, C++, and Java use braces { } to define a block of code. Python uses indentation. A code block (body of a function, loop, etc.) starts with indentation.

Eg:

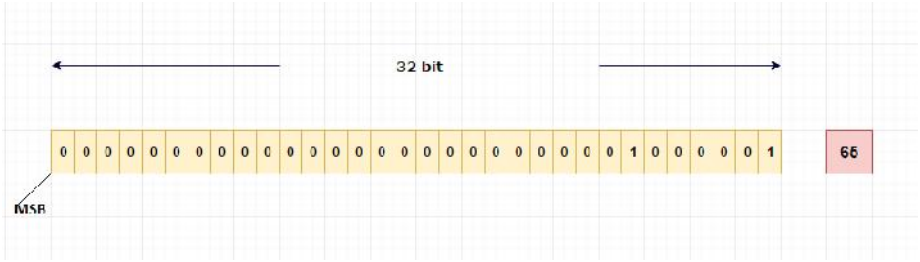
```
for i in range(1,11):
    print(i)
    if i == 5:
        break
```

Python Data Types: int, float, str, complex, set, tuple, bool etc….

Print the data type of the variable x: **print(type(x))**. In Python, the data type is set when you assign a value to a variable. If you want to specify the data type, you can use Eg: x = int (20), y = float (input (…..)) etc…

Python Operators

1. **Arithmetic: +, -, *, /, %, **, //**
2. **Logical :** and, or, not
3. **Identity:** is, is not
4. **Membership :** in, not in
5. **Assignment: =, +=, -=, *=, /=, %=, //=, **=**
6. **Comparison: ==, !=, >, <, >=, <=**
7. **Conditional (Ternary):** [on_true] if [expression] else [on_false]
   Eg: out = a if a < b else b
8. **Bitwise:** &,|,^,~,<<,>>
   Eg:
   Binary to decimal




```
a = 60            # 60 = 0011 1100
b = 13            # 13 = 0000 1101
c = 0
c = a & b;        # 12 = 0000 1100
c = a | b;        # 61 = 0011 1101
c = a ^ b;        # 49 = 0011 0001
c = ~a;           # -61 = 1100 0011(negative numbers Two’s compliment representation)
              # (Take 61 binary =>0011 1101 => negate=> 1100 0010(ones complement)=>
add 1=>1100 0011(twos complement))

c = a << 2;       # 240 = 1111 0000
c = a >> 2;       # 15 = 0000 1111
```

Lecture 5

Programming Exercises:

1) Check the given number (via keyboard) is divisible by 6 or not?
2) Print the multiplication table (1 to 10) for the given number (1 to 10)?
3) Count how many even digits are there for the given number?
4) Print the reversal of the given number?
5) Write a program to print the decimal number representation for the given binary?
6) Print the following pattern for the given number of rows?

```
1
*  +
3  6  9
*  +   *  +
5  10  15  20 25
*  +   *  +  *   +
```

Do these programs and submit to me at praveen.cys@gmail.com