Python Assignment_1

1)

- --> mathematical operator

'hello' --> string

-87.8 --> flot

- --> mathematical operator

/ --> mathematical operator

- --> mathematical operator

6 --> int

2)

Variables stores different values of different data types.

String is a data type.

3)

Three data types in python are:

i) **INT** --> Int data type stores integer or numeric (except decimal point)

eg: -2, 3, 6, 8

ii) **STRING** --> String data type stores non-numeric values like letters, words, alphabets, and sentences.

eg: "My name is Ananya Dhar"

iii) **Float** --> Floating point number are those with point. Numeric values with decimal points.

eg: 3.5

4)

An expression is a contains variables, values, operators and operands and functions to give a result.

Expressions have all the required information like values, operators, operands and functions to give the desired results.

eg: a = 2

b = 2

c = a + b # this is an expression

5)

Statement --> a statement declares the value of a variable

eg: a = 5 is a statement

Expression --> an expression is formed when statements are operated upon with various operators to give a result

eg: a =5

b = 3

c = a + b is an expression

6)

The variable bacon contains the numeric value of 22. After incrementing bacon by 1, the value becomes 23. However if we print(bacon), the output is 22.

7)

```
In [11]: print('spam'+'spamspam')
         spamspamspam
In [12]: print('spam'*3)
         spamspamspam
```

8)

There are certain variable conventions in python: 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)

In case of eggs, it abides by all the rules. But since a variable name cannot start with a number, 100 is illegal.

9)

Integer --> int()

Floating --> float()

String --> str()

10)

The following is an example of string concatenation. We can concat two strings but can't concat string to integer. Hence the compiler throws an error.

If we insert 99 as "99" and then concat it, the compiler does not throw an error.

'i have eaten'+'99'+'burritos.'

```
In [ ]:
```