**Assignment\_1**

1. In the below elements which of them are values or an expression? eg:- values can be integer or string and expressions will be mathematical operators.

\*

'hello'

-87.8

-

/

6

**Answer**: Values : ‘hello’, -87.8, 6

Expression: \*, -, /, +

2. What is the difference between string and variable?

**Answer:** A **string** is a datatype for a sequence of one or more characters (letters, numbers,

symbols) that can be either a constant or a variable.

A **variable** is a symbolic name that is a reference or pointer to an object.

Once an object is assigned to a variable, you can refer to the object by that name.

3. Describe three different data types.

**Answer:** **Integers** are one of the types that can be used to represent numbers in Python.

These can be both positive and negative values.

**Boolean** is a built-in logical data type that is mainly used for checking whether the

logic of an expression or comparison is true or not. Boolean variables can only have

two possible values: True or False.

**Floats** are another form of representing numbers in Python and exist as either

positive or negative decimal values.

4. What is an expression made up of? What do all expressions do?

**Answer:**

An **expression** is a combination of operators and operands that is interpreted to produce some other value. It is evaluated as per the precedence of its operators.

**Constant Expressions**: These are the expressions that have constant values only.

Ex: x = 5 +4

**Arithmetic Expressions:** An arithmetic expression is a combination of numeric values, operators, and sometimes parenthesis. The result of this type of expression is also a numeric value.

Ex: x = 40

y = 12

add = x + y

**Integral Expressions:** These are the kind of expressions that produce only integer results after all computations and type conversions.

Ex: a = 13

b = 12.0

c = a + int(b)

**Floating Expressions:** These are the kind of expressions which produce floating point numbers as result after all computations and type conversions.

Ex: a = 13

b = 5

c = a / b

**Relational Expressions:** In these types of expressions, arithmetic expressions are written on both sides of relational operator (> , < , >= , <=). Those arithmetic expressions are evaluated first, and then compared as per relational operator and produce a boolean output in the end.

a = 21

b = 13

c = 40

d = 37

p = (a + b) >= (c - d)

**Logical Expressions:** These are kinds of expressions that result in either True or False. It basically specifies one or more conditions.

Ex: P = (10 == 9)

Q = (7 > 5)

R = P and Q

**Bitwise Expressions:** These are the kind of expressions in which computations are performed at bit level.

Ex: a = 12

x = a >> 2

y = a << 1

**Combinational Expressions:** We can also use different types of expressions in a single expression, and that will be termed as combinational expressions.

Ex: a = 16

b = 12

c = a + (b >> 1)

5. This assignment statements, like spam = 10. What is the difference between an expression and a statement?

**Answer:** A statement is a complete line of code that performs some action, while an expression is any section of the code that evaluates to a value. Expressions can be combined “horizontally” into larger expressions using operators, while statements can only be combined “vertically” by writing one after another, or with block constructs. Every expression can be used as a statement (whose effect is to evaluate the expression and ignore the resulting value), but most statements cannot be used as expressions.

Expression Ex : 2 + 2

Statement Ex: If Condition, Elif condition etc.

6. After running the following code, what does the variable bacon contain?

bacon = 22

bacon + 1

**Answer:** Variable “bacon” will have the value 22 even though we are incrementing the value by 1 because while incrementing value of bacon we are not again re-initialising it with the new value of bacon.

7. What should the values of the following two terms be?

'spam' + 'spamspam'

'spam' \* 3

**Answer:** Both will return the same output, i.e. ‘spamspamspam’. First example is having concatenate operator which concatenates two strings, whereas Second example is having string repeating operator which repeats the string three times and presented the output.

8. Why is eggs a valid variable name while 100 is invalid?

**Answer:** “eggs” is variable with name “eggs” which is a container itself and is not a value itself. However 100 is literal number having value as 100. We cannot assign a value to a literal value such as a boolean, a string, a list, or a number.

9. What three functions can be used to get the integer, floating-point number, or string version of a value?

**Answer:** int function to get integer version : x = int(“100”)

float function to get floating-point number version : y = float(100)

str function to string version : x = str(100)

10. Why does this expression cause an error? How can you fix it?

'I have eaten ' + 99 + ' burritos.'

**Answer:** In the above question, we are trying to concatenate string and integer and that’s why its throwing error. Error can be rectified by converting integer to string while printing.

Ex: ‘I have eated ‘+ str(99) + ‘ burritos’