# Assignment 1:

## Q1.

**Values:**

'hello' (string value)

-87.8 (floating-point value)

**Expressions:**

\* (multiplication operator)

- (subtraction operator)

/ (division operator)

+ (addition operator)

## Q2.

**String:**

A string represents a sequence of characters. Like an array.

Strings are enclosed in quotation marks. [“ “]

**Variable:**

A variable is used to assign a value. Variables are used to store and manipulate.

A variable can hold different types of values, including strings, numbers, or other data types.

Variables are typically assigned a value using an assignment operator (=).

## Q3.

**Integer (int):**

The integer data type represents whole numbers without any fractional part.

Examples : -5, 0, 42, 1000

Integers can be used in mathematical calculations and logical operations.

**String (str):**

The string data type represents a sequence of characters.

Strings are enclosed in quotation marks (single or double).

Examples : "Hello", 'Python', "42"

**Float:**

The float data type represents decimal numbers or numbers with fractional parts.

Floats are used when more precision is needed in numerical calculations or when representing real-world values with decimal places.

Example: 3.14

## Q4

An expression is made up of different components that work together to produce a value. It consists of variables, values, and operators.

Variables: Variables are like containers that hold values. They can store different types of data such as numbers or text.

Values: Values are the actual data that the expression operates on. They can be numbers (like 5 or 3.14) or text (like "Hello" or "Python").

Operators: Operators are symbols that perform operations on variables and values.

Expressions, when evaluated, produce a single value as the result. They can perform mathematical calculations or carry out logical operations.

## Q5

Expression:

An expression is a combination of variables, values, and operators that evaluates to a single value.

It produces a result or value when it is executed.

Examples of expressions: 2 + 3, x \* y, "Hello" + "World"

Statement:

A statement is a complete instruction or action that performs a specific task.

It represents a line or block of code that performs an action or controls the flow of a program.

Examples of statements: if statements, for loops, function definitions.

## Q6

After running the given code, the variable bacon would still contain its original value, which is 22. It doesn't modify the existing value.

## Q7

The values of the two terms would be:

'spam' + 'spamspam':

The resulting value is 'spamspamspam'.

'spam' \* 3:

The resulting value is 'spamspamspam'.

## Q8

In Python, variable names must follow certain rules to be considered valid.

"eggs" is a valid variable name because it starts with a letter and only contains valid characters (letters in this case).

"100" is an invalid variable name because it starts with a digit (1). Variable names cannot start with a digit in Python.

## Q9

To convert a value to different data types in Python, we can use the following three functions:

int():

The int() function is used to convert a value into an integer data type.

float():

The float() function is used to convert a value into a floating-point number data type.

str():

The str() function is used to convert a value into a string data type.

## Q10

The expression 'I have eaten ' + 99 + ' burritos.' causes an error because it attempts to concatenate a string with an integer value directly, which is not allowed in Python. The + operator is used for string concatenation when both operands are strings. However, in this case, the value 99 is an integer, not a string.

To fix the error and perform the concatenation correctly, we need to convert the integer value to a string before concatenating it with the other strings. we can achieve this by using the str() function to convert the integer to a string. Here's the corrected expression:

'I have eaten ' + str(99) + ' burritos.'