Assignment No-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

Solution: In the given elements:

- 'hello' is a value, as it is a string.

- -87.8 is a value, as it is a negative floating-point number.

- - is an expression, as it is a mathematical operator for subtraction.

- / is an expression, as it is a mathematical operator for division.

- + is an expression, as it is a mathematical operator for addition.

- 6 is a value, as it is a positive integer.

2. What is the difference between string and variable?

Solution: A string is a sequence of characters in quotation marks (e.g., "Hello, World!"). It represents textual data in programming.

On the other hand, a variable is a named container that holds a value. It has a memory address where the value is stored. Variables can hold various types of data, including strings, numbers, booleans, etc.

In simpler terms, a string is a data type, whereas a variable is a container that can hold different data types, including strings. A variable can store and manipulate strings and other data types, like integers or floating-point numbers.

3. Describe three different data types.

Solution: a) Integer: The integer data type represents whole numbers. It is used to store values that do not have decimal places and can be positive, negative, or zero. For example, the number 5 and -10 are integers.

b) Float: Float data type represents decimal numbers. It is used to store values with a fractional part. Floats can have varying precision and can be positive or negative. For example, 3.14 and -2.5 are float numbers.

c) String: String data type represents a sequence of characters. It is used to store alphanumeric or textual data. Strings are enclosed in quotation marks, either single (' ') or double (" "). For example, "Hello, World!" and "123abc" are examples of strings. Strings can store text, names, addresses, and more.

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

Solution: An expression is made up of variables, constants, and operators. It represents a value or a computation.

All expressions are evaluated to a value. They can perform calculations, assign variable values, or provide control structure conditions. Expressions can also be combined to form more complex expressions or used as arguments in functions.

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

Solution: An expression is a piece of code that evaluates to a value, whereas a statement is a line of code that performs an action.

In the example "spam = 10", this is a statement where the value 10 is assigned to the variable spam. This statement doesn't produce a value by itself, but it has the side effect of changing the value of the variable spam.

However, if we have an expression like "spam + 5", this will evaluate to a value (in this case, 15) because it performs a computation. Expressions can be used within statements, for example, in an if statement where the result of an expression determines the flow of the program.

In summary, an expression produces a value, while a statement is an action that may have side effects and does not necessarily produce a value.

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

bacon = 22

bacon + 1

Solution: The variable "bacon" still contains the value 22. The code's expression "bacon + 1" does not assign the result back to the variable "bacon". To update the value of "bacon", you need to assign the expression back to the variable: "bacon = bacon + 1".

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

'spam' + 'spamspam'

'spam' * 3

Solution: The values of the following two terms are:

a) 'spam' + 'spamspam': The value will be 'spamspamspam'. This is the concatenation of the two strings 'spam' and 'spamspam', resulting in a new string 'spamspamspam'.
b) 'spam' * 3: The value will be 'spamspamspam'. This is achieved by multiplying the string 'spam' by the number 3, which repeats the string three times to form a new string, 'spamspamspam'.

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

Solution: In most programming languages, variable names follow specific rules and conventions. One of the rules is that variable names cannot start with a number. They can start with a letter, an underscore, or a unique character (depending on the programming language).

Therefore, "eggs" is a valid variable name because it starts with a letter. On the other hand, "100" is an invalid variable name because it starts with a number.

It's important to use valid variable names to follow the programming language's rules and make the code more readable and understandable.

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

Solution: The three functions that can be used to get the integer, floating-point number, or string version of a value are as follows:

a) ‘int()’: This function can convert a value into an integer. If possible, it returns the integer version of a value, truncating any decimal places. For example, ‘int(3.14)’ will return ‘3’.
b) ‘float()’: This function can convert a value into a floating-point number. It returns the floating-point version of a value if possible. For example, ‘float("3.14")’ will return ‘3.14’.
c) ‘str()’: This function can convert a value into a string. It returns the string version of a value, representing it as a sequence of characters. For example, ‘str(42)’ will return ‘'42'’.

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

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

Solution: This expression causes an error because it is trying to concatenate a string with a number without converting the number to a string first.

To fix it, you can convert the number to a string before concatenating it by using the str() function. Here is the corrected expression:

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