**Assignment 1**

**Q1. 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**

**Ans:** The Below elements are as follows:

* **\*:** expression (multiple operator)
* ‘**hello’**: value (string)
* **-87.8**: value (float)
* **-**: expression (subtraction operator)
* **/**: expression (division operator)
* **+**: expression (addition operator)
* **6**: value (integer)

**Q2. What is the difference between string and variable?**

**Ans:** A string is a sequence of characters enclosed within quotation marks (either single or double quotes) in Python. It represents textual data and is immutable, which means that once a string object is created, its contents cannot be modified.

A variable is a named location in memory that holds a value of a certain data type, such as string, integer, float, or Boolean. Variables are used to store and manipulate data in a program. They can be assigned a value using the assignment operator (=) and can be modified by assigning a new value to them.

**Q3. Describe three different data types.**

**Ans:** The three of the most commonly used data types are:

1. Integer (int): An integer is a whole number, meaning it has no decimal places. Integers are represented using the **‘int’** type. For example: 5, 100, -2, 0 are all integers.
2. Boolean: A Boolean is a data type that can have only two possible values: ‘True’ or ‘False’. The **‘bool’** type is used to represent Boolean values.
3. String: A string is a sequence of characters enclosed in (single or double) quotation marks. Strings in Python are represented using the **‘str’** type. For example: “hello”, “World”, “123” are all strings.

**Q4. What is an expression made up of? What do all expressions do?**

**Ans:** An expression is made up of values, operators, and function calls that can be evaluated to produce a value. An expression can consist of a single value, such as an integer or a string, or it can be more complex and involve multiple values and operators.

Expressions can perform a wide range of operations in Python, such as arithmetic calculations, logical operations, string manipulations, and more. When an expression is evaluated, it produces a value of a certain data type. For example, an arithmetic expression like ‘**2 + 3**’ would evaluate to the integer value ‘**5’**.

All expressions have the same basic purpose, which is to compute a value. Expressions can be used in a variety of ways in a program, such as to assign values to variables, to compare values, or to control the flow of a program using conditional statements and loops.

**Q5. This assignment statements, like spam = 10. What is the difference between an expression and a statement?**

**Ans:** An expression is a combination of values, operators, and function calls that can be evaluated to produce a value. Expressions always return a value when they are evaluated, and they can be used as part of a larger expression or as an argument to a function.

A statement is a unit of code that performs some action but does not necessarily return a value. Statements can perform a wide range of actions in Python, such as assigning values to variables, calling functions, looping over data, or making decisions using conditional statements.

The assignment statements ‘**spam** **= 10**’ is an example of a statement. It assigns the value 10 to the variable ‘**spam**’ and does not return any value.

The main difference between expressions and statements is that expressions always return a value when they are evaluated, while statements perform some action but may not necessarily return a value.

**Q6. After running the following code, what does the variable bacon contain?**

**bacon = 22**

**bacon + 1**

**Ans:** After running the code, the variable **bacon** would contain the value **22.**

The **bacon +1** code evaluates the expression **bacon + 1,** which adds 1 to the value **bacon** (which is 22). However, this expression does not assign the result to any variable, so the value of **bacon** remains unchanged.

**Q7. What should the values of the following two terms be?**

**‘spam’ + ‘spamspam’**

**‘spam’ \* 3**

**Ans:** The values of the terms would be:

1. ‘spam’ + ‘spamspam’ evaluates to the string ‘spamspamspam’. This is because the ‘+’ operator performs concatenation on two strings, joining them together to create a new string that contains the characters of both strings in order.
2. ‘spam’ \*3 evaluates to the string ‘spamspamspam’. This is because the ‘\*’ operator performs string repetition, repeating a string a certain number of times. In this case, ‘spam’ \* 3 repeats the string ‘spam’ three times, resulting in the string ‘spamspamspam’.

**Q8. Why is eggs a valid variable name while 100 is invalid?**

**Ans:** A variable names must follow certain rules in order to be valid. Specially, variable names:

* Must start with a letter (either uppercase or lowercase) or an underscore.
* May contain letters, numbers, and underscores.
* Cannot start with a number.

Based on these rules, ‘eggs’ is a valid variable name because it starts with a letter and contains only letters and a lowercase letter. In contrast, ‘100’ is an invalid variable name because it starts with a number, violating the third rule for valid variable names.

**Q9. What three functions can be used to get the integer, floating-point number, or string version of a value?**

**Ans:** The three functions that can be used to convert a value to an integer, floating-point number, or string are:

1. **int ()**: This function can be used to convert a value to an integer. It takes one argument, which is the value to be converted. If the value is a floating-point number or a string containing an integer, it will be converted to an integer. If the value is not numeric, a ‘**ValueError’** will be raised.
2. **float ():** This function can be used to convert a value to aa floating-point number. It takes one argument, which is the value to be converted. If the value is an integer or a string containing a floating-point number, it will be converted to a float. If the value is not numeric, a ‘**ValueError’** will be raised.
3. **str ():** This function can be used to convert a value to a string. It takes one argument, which is the value to be converted. The function will return a string representation of the value.

**Q10. Why does this expression cause an error? How can you fix it?**

**‘I have eaten’ + 99 + ‘burritos.’**

**Ans:** This expression causes an error because we cannot concatenate a string with a number directly. The ‘+’ operator can be used to concatenate two strings, but it cannot concatenate a string and a number.

To fix the error, we need to convert the number to a string before concatenating it with the other strings. We can use the ‘**str ()**’ function to convert the number to a string.

We can fix the expression by writing: ‘**I have eaten’ + str (99) + ‘burritos.**’