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.

\* = multiplication

'hello' = string

-87.8 = Float

- = subtraction

/ = Division

* = Addition

6 = integer

2. What is the difference between string and variable?

Ans.2) A variable is a named storage location in a computer's memory that holds a value. whereas, A string is a sequence of characters enclosed in quotation marks. It represents textual data in programming languages.

3. Describe three different data types.

Ans.3) a) The integer data type represents whole numbers without any decimal points. It includes both positive and negative numbers, as well as zero. In most programming languages, integers have a fixed size, which determines the range of values they can hold.

b) The string data type represents a sequence of characters. It is used to store textual data, such as names, sentences, or any other combination of letters, digits, symbols, and white spaces.

C) The Boolean data type represents logical values that can be either true or false. Booleans are commonly used in programming for making decisions and controlling the flow of a program

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

Ans.4) An expression is a combination of values, variables, operators, and function calls that evaluate to a result. It can be as simple as a single value or as complex as a combination of multiple sub-expressions.

1.Values: These are literal data elements such as numbers, strings, or boolean values. Examples include 42, "Hello", or True.

2.Variables: Variables are symbols that represent values stored in memory. They can be assigned values and used in expressions. For example, x or my\_variable could be variables.

3.Operators: Operators perform operations on one or more values or variables to produce a result. They can be arithmetic operators (e.g., +, -, \*, /), comparison operators (e.g., ==, >, <, !=), logical operators (e.g., and, or, not), and more.

4.Function calls: Functions are reusable blocks of code that perform specific tasks. Function calls invoke these functions with specified arguments. Functions can also be part of expressions, and their return values can be used in computations.

Expressions are used to compute values, make comparisons, or perform other operations. When an expression is evaluated, it produces a value based on the combination of its components.

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

Ans.5) An expression is a combination of variables, values, and operators that produces a single value when evaluated. It can be as simple as a single constant value or a complex combination of operations. Examples of expressions include numeric calculations, string concatenation, function calls, and logical comparisons. Expressions can be used within statements or as standalone entities whereas A statement is a complete instruction that performs an action or control flow within a program. It can consist of one or more expressions, along with keywords, variables, and operators, to execute specific tasks. Statements are the building blocks of programs and define the program's behavior. Examples of statements include variable assignments, control flow statements (if-else, loops), function definitions, and function calls.

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

bacon = 22

bacon + 1

Ans6.) After running the given code, the variable "bacon" will still contain the value 22.

Here's the breakdown:

bacon = 22: This line assigns the value 22 to the variable bacon. So, bacon is now equal to 22.

bacon + 1: This line calculates the expression bacon + 1, which evaluates to 23. However, this expression is not assigned to any variable or used in any way, so it doesn't affect the value of bacon.Therefore, the variable bacon remains unchanged and still contains the value 22.

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

'spam' + 'spamspam'

'spam' \* 3

Ans7.)The values of the two terms would be:

'spam' + 'spamspam': This concatenates the string 'spam' with the string 'spamspam', resulting in the value 'spamspamspam'.

'spam' \* 3: This multiplies the string 'spam' by 3, which repeats the string three times, resulting in the value 'spamspamspam'.

In both cases, the resulting value would be the string 'spamspamspam'.

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

Ans8.)

In most programming languages, including Python, variable names need to follow certain rules and conventions. Here's why 'eggs' is a valid variable name while '100' is invalid:

Valid variable names: Variable names can consist of letters (both uppercase and lowercase), digits, and underscores.

However, a variable name cannot start with a digit. It must begin with a letter or an underscore.

Explanation: 'eggs': This variable name starts with a letter ('e') and contains only letters. Therefore, it satisfies the rules for valid variable names.

'100': This variable name starts with a digit ('1'), violating the rule that variable names cannot begin with a digit. Therefore, it is not a valid variable name.

It's worth noting that adhering to good naming conventions is also important for code readability and maintainability. Variable names should be descriptive and meaningful, helping to make the code more understandable to other developers and yourself in the future

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

Ans9.) In Python, you can use the following three functions to convert a value to an integer, floating-point number, or string:

1.) Int (): This function is used to convert a value to an integer. It takes a numeric or string argument and returns an integer representation of that value. If the argument is a floating-point number, it will be truncated towards zero.

2.) Float (): This function is used to convert a value to a floating-point number. It takes a numeric or string argument and returns a floating-point representation of that value.

3.) Str (): This function is used to convert a value to a string. It takes any object as an argument and returns a string representation of that object.

These functions provide convenient ways to convert values between different data types, allowing you to work with them in the desired format within your Python code.

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

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

Ans10.) The expression 'I have eaten ' + 99 + ' burritos.' causes an error because it tries to concatenate a string ('I have eaten ') with an integer (99) directly without converting the integer to a string. In Python, you cannot concatenate different data types together without explicitly converting them to a compatible type.

To fix the error and perform the concatenation correctly, you need to ensure that all elements being concatenated are of the same data type. In this case, you can convert the integer 99 to a string before concatenating it with the other strings. Here's the corrected

In the corrected version, the str (99) converts the integer 99 to a string, allowing it to be concatenated properly with the other strings. This expression will result in the desired output: 'I have eaten 99 burritos.'