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

Ans: values=‘hello’,-87.8,6 , mathematical operators=\*,-,/,+

2. What is the difference between string and variable?

Ans: A string is a sequence of characters enclosed within either single (' ') or double (" ") quotation marks. Strings are used to represent textual data

And

A variable is a name that refers to a value or an object in memory. It acts as a placeholder to store and manipulate data. Variables can hold various types of data, including strings. They provide a way to label and reference values.

3. Describe three different data types.

**Ans:1.Integer (int):**

Integers are whole numbers, both positive and negative, without any fractional part. They are used to represent discrete quantities in programming.

**2.Float (float):**

Floats, short for floating-point numbers, are used to represent numbers with a fractional part. They can also represent very large or very small values using scientific notation.

**3.String (str):**

Strings are sequences of characters enclosed within quotation marks. They are used to represent textual data. Strings can contain letters, numbers, symbols, and spaces.

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

Ans: An expression in Python is a combination of values, variables, operators, and function calls that can be evaluated to produce a result. Expressions are the building blocks of Python code and are used to perform computations, create data structures, and make decisions. Expressions can be as simple as a single value or can involve complex operations.

All expressions in Python serve the purpose of producing a value or result when they are evaluated. This value can be of various types, such as numbers, strings, boolean values, or more complex data structures like lists or dictionaries.

5. This assignment statements, like spam = 10. What is the difference between an

expression and a statement?

Ans: **Expression:**

An expression is a combination of values, variables, operators, and function calls that can be evaluated to produce a value. Expressions are meant to compute values and are often used within larger constructs like statements. Expressions can be as simple as a single value or as complex as a combination of multiple operations.

Examples of expressions:

5 + 3

x \* 10

len("hello")

Expressions are designed to be evaluated and return a result but don't have inherent side effects.

**Statement:**

A statement is a complete line of code that performs an action. Statements are used to control the flow of a program, execute instructions, or define structures. Unlike expressions, statements often do something (change a variable, perform an action, define a function or class) rather than just calculating a value.

Examples of statements:

Assignment statement: x = 10

Conditional statement: if x > 5:

Loop statement: for item in list:

Function definition: def my\_function():

Statements can have side effects, like modifying variables, changing the state of the program, or affecting its behavior.

In summary, the main difference between expressions and statements is that expressions focus on producing values, while statements focus on performing actions or controlling the program's flow. Additionally, expressions can often be part of statements, such as using an expression within an assignment statement.

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

bacon = 22

bacon + 1

Ans:22, as 22 was assigned to the variable “bacon” and on the later line 1 id added to the value of bacon but the value assigned to bacon has not changed

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

‘spam’ + ‘spamspam’

‘spam’ \* 3

Ans: Following is the cde and output from python

"spam"+"spamspam"

'spamspamspam'

"spam"\*3

'spamspamspam'

Which indicates both the values will be the same

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

Ans: the reason "eggs" is a valid variable name while "100" is not comes down to adhering to the rules for variable naming in Python. Variable names must start with a letter or underscore, cannot start with a digit, and can only contain letters, digits, and underscores after the initial character.

9. What three functions can be used to get the integer, floating-point number, or string

version of a value?

Ans: ‘int()’,’float()’ and ‘str()’ respectively

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

‘I have eaten ‘ + 99 + ‘ burritos’

Ans: The expression 'I have eaten ' + 99 + ' burritos' causes an error because you are trying to concatenate a string with an integer without explicitly converting the integer to a string. In Python, can't directly concatenate a string with an integer using the + operator.

To fix this error, need to convert the integer 99 to a string before concatenating it with the other strings. Here's the corrected version of the expression:

Python code:'I have eaten ' + str(99) + ' burritos'

In this corrected version, the str(99) function call converts the integer 99 to its string representation, allowing you to concatenate it with the other strings using the + operator without any errors.