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

2. What is the difference between string and variable?

3. Describe three different data types.

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

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

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

bacon = 22

bacon + 1

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

'spam' + 'spamspam'

'spam' * 3

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

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

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

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

Q1. values=-87.8,6,'hello' expressions=*,-,/,-

Q2. What is the difference between string and variable? Variables are symbols that we can use to store data in a program. We can think of them as an empty box that we fill with some data or value. Strings are data, so we can use them to fill up a variable. Declaring strings as variables can make it easier for us to work with strings throughout our Python programs.

Q3. Describe three different data types. Variables can store data of different types, and different types can do different things.

1.Text Type : str 2.Numeric Types : int, float, complex 3.Sequence Types: list, tuple, range 4.Mapping Type : dict 5.Set Types : set, frozenset 6.Boolean Type : bool 7.Binary Types : bytes, bytearray, memoryview 8.None Type : NoneType

Example Data Type x = "Hello World" str x = 20 int x = 20.5 float x = 1j complex x =["apple", "banana", "cherry"] list x = ("apple", "banana", "cherry") tuple x = range(6) range x = {"name" : "John", "age" : 36} dict x = {"apple", "banana", "cherry"} set x = frozenset({"apple", "banana", "cherry"}) frozenset x = True bool x = b"Hello" bytes x = bytearray(5) bytearray x = memoryview(bytes(5)) memoryview x = None NoneType

List Items List items are ordered, changeable, and allow duplicate values. thislist = ["apple", "banana", "cherry", "apple", "cherry"]

Tuple Items Tuple items are ordered, unchangeable, and allow duplicate values. thistuple = ("apple", "banana", "cherry", "apple", "cherry")

Set A set is a collection which is unordered, unchangeable*, and unindexed. thisset = {"apple", "banana", "cherry"}

Dictionary Items Dictionary items are ordered, changeable, and does not allow duplicates. Dictionary items are presented in key:value pairs, and can be referred to by using the key name. thisdict = { "brand": "Ford", "model": "Mustang", "year": 1964 }

Q4. What is an expression made up of? What do all expressions do? An expression is a combination of operators and operands that is interpreted to produce some other value. In any programming language, an expression is evaluated as per the precedence of its operators.

5.This assignment statements, like spam = 10. What is the difference between an expression and a statement? An expression can be defined as any element in our utility that evaluates some values. An expression evaluates a value.

Statements represent a motion or command e.g print statements, mission statements. Expressions need to be evaluated.

An expression is any series of symbols that represents a value. For example, 2.9, a + 5, (3 + 5) / 4.

For example: print("Hello")

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

bacon = 22 bacon + 1 Ans. 22

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

'spam' + 'spamspam'

'spam' * 3 Ans.spamspamspam

Q8.Why is eggs a valid variable name while 100 is invalid? Ans. because varible cannot start with number and other special charcter it only be start with letter or underscore

Q9. What three functions can be used to get the integer, floating-point number, or string version of a value? Ans. int() float() str()

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

'I have eaten ' + 99 + ' burritos.' Ans. 1.print(f"I have eaten {99 } burritos.") 2.print("I have eaten", 99,"burritos.")

In [ ]: