

Module 2

Python Assignment

4> How is memory managed in Python?

Ans. In Python, memory management is handled automatically by the interpreter. It has a built-in garbage collector that automatically frees up memory that is no longer being used by the program. When an object is no longer being referenced by any part of the program, the garbage collector identifies it and frees up the memory it was occupying. The garbage collector runs periodically and is responsible for reclaiming memory that is no longer needed. Python also uses a technique called reference counting to keep track of objects in memory. Each object has a reference count that indicates how many references to the object exist in the program. When the reference count for an object drops to zero, the garbage collector will free up the memory occupied by the object. Overall, Python's memory management system is designed to be efficient and automatic, so that developers don't have to worry about managing memory directly.

5> What is the purpose of continue statement in python?

Ans. The continue statement is used to skip the current iteration of a loop and move on to the next iteration. When the continue statement is encountered in a loop, the current iteration of the loop is terminated, and the next iteration begins immediately.

14> What are negative indexes and why are they used?

Ans. indexes are used in arrays in all the programming languages.

-> We can access the elements of an array by going through their indexes.

-> But no programming language allows us to use a negative index value such as -4.

-> Python programming language supports negative indexing of arrays, something which is not available in arrays in most other programming languages.

-> This means that the index value of -1 gives the last element, and -2 gives the second last element of an array.

-> The negative indexing starts from where the array ends.

-> This means that the last element of the array is the first element in the negative indexing which is -1.

Example:

```
arr          =          [10,          20,          30,          40,          50]
print                                              (arr[-1])
print (arr[-2])
```

Output:

50

40