MEMORY MANAGEMENT IN PYTHON

Memory management is the process of controlling and coordinating the use of memory in a computer system. This includes allocating memory to different programs and processes, making sure that the memory is used efficiently, and freeing up memory that is no longer needed.

The goal of memory management is to ensure that the computer system runs smoothly and efficiently by using memory resources effectively.

Memory Management in Python:

In Python, memory management is the process by which the Python interpreter manages the memory used by the program. Python uses a private heap space to store all its objects and data structures. The memory manager in Python is responsible for allocating and deallocating memory for these objects as needed, and for keeping track of which parts of the memory are in use and which are available for use. When a variable is assigned a value, the memory manager will allocate a block of memory to store the value. When the variable is no longer needed, the memory manager will automatically free up that block of memory. This allows Python to manage the memory used by the program automatically, so that the programmer doesn't have to worry about manually allocating and freeing memory.

This means that the memory manager keeps track of the number of references to each object in the program. When an object’s reference counts drops to zero, which means the object is no longer being used, the garbage collector (part of memory manager) automatically frees the memory from that particular region.

There are two types of memory:

1)STACK (STATIC MEMORY):The program is allocated memory at compile time. The memory is allocated at the time of compilation. Stack is used for implementing static compilaton. In this case memory cannot be reused.

Static int a=10;

2)HEAP (DYNAMIC MEMORY ALLOCATION):The program is allocated memory at runtime.The memory is allocated at runtime. Heap is used for implementing dynamic allocation. In this case, memory can be freed and reused when not required.

Int \*p;

P=new int;

Everything in python is an object. This means that Dynamic Memory Allocation underlies Python Memory Management. When objects are no longer needed, the Python Memory Manager will automatically reclaim memory from them.

PYTHON GARBAGE COLLECTION

The Python garbage collector is a system that automatically manages the memory used in python program. It keeps track of all the objects that are created by the program and periodically frees up memory by removing objects that are no longer being used. This helps to prevent the program from using too much memory and causing the system to run out of resources.

PYTHON OBJECTS IN MEMORY

In Python , everything is an object, including variables, dictionaries, and functions. When you create a variable or an object, Python sets aside a space in the computer’s memory to store it. This memory space can be accessed and manipulated by the program as long as the objects remains alive , meaning it is still being referenced by the program.

Objects can either be simple or containers. Furthermore, Python is a dynamically typed language which means that we do not need to declare the variable or their types before using them in a program.

PYMALLOC IN PYTHON

“Pymalloc” is an official memory manager that can be used to improve the performance of python’s build-in memory management. It is a custom memory allocator that is optimized for allocating small objects, like those used in python. It uses a technique called “pool allocation” to quickly allocate and free small blocks of memory, which can help reduse memory fragmentation and increase memory allocation performance.

REFERENCE COUNTING IN PYTHON:

It is a technique used to keep track of reference to an object in memory. When an object is created, its reference count is set to 1. When a reference to the object is stored in a different object, its reference count is incremented by 1. Similarly, when a reference count is deleted,its reference count is decremented by 1. When an object’s reference count reaches to zero , the objects memory is deallocated and the object can no longer be used.