**Python**

**1) How is Memory is managed in Python?**

Python Memory management is handled by Private heap space.

The heap is the region of memory where objects are stored and managed.

Python Memory manager takes care of allocating memory to new objects and freeing up the memory for unused objects.

In Python, Memory management is automatic unlike in C or CPP

In python reference count is the technique to track the objects in the memory.

It is incremented when new references for the object is created. and decrmented when the reference are deleted. When the reference count reaches zero then the memory for the object is freed up.

- Reference count technique has it's own limitations. It can't detect reference cycles. To solve this python uses garbage collector.

- Python's garbage collector checks the all the objects that are not reachable from the root of the object graph and frees up the memory occupied by these objects.

- The garbage collector uses the Mark and Sweep algorithm. It marks the objects which are reachable from the root and freeing up the memory occupied by unmarked objects.

- Garbage collector runs in background and runs whenever it's necessary.

#To get reference count

sys.getrefcount(a)

# To access the garbage collector to release the memory explicitly

import gc

gc.collect()

It makes significant difference when huge number of objects are deallocated the memory.

**2) How does Multithreading work in Python?**

The program or app that executes is called as process. Each process can have multiple tasks that run concurrently.

Thread is nothing but a independent flow of execution. It can also be defined as instance of a process.

Threading module can be used to implement Multithreading in python.

GIL is type of process when it is working with multiple processes.

GIL is used in python to lock the thread during it’s processing.

The main concern with reference count variable is that can be effected when two or three threads trying to increment or decrement the reference count value. It is called race condition.

If this condition happens it causes leaked memory that is never released.

GIL avoids this condition by locking the threads.

Multiprocessing method is used to deal with GIL in python. But multiprocess has it’s own overhead, multi processes are heavier than multi threads.

Multiple threads can be run using multiple processors.

**3) What is Monkey Patching in Python?**

Monkey patching is the technique of dynamic modification of piece of code during runtime without affecting the source code.

- It is useful to modify the code when the we don't have the access to the original source code to change.

#monk.py

**class** A:

**def** func(self):

          print ("func() is being called")

#sample1.py

**import** monk

**def** monkey\_f(self):

**print** ("monkey\_f() is being called")

# replacing address of "func" with "monkey\_f"

monk.A.func **=** monkey\_f

obj **=** monk.A()

# calling function "func" whose address got replaced

# with function "monkey\_f()"

obj.func()

#Output would be

monkey\_f() is being called.

4) What is Type hinting in python?

Type hints indicate the type of variables, function parameters and return values.

age : int = 20

name : str ='Hello'

def greet(name:str) -> str:

4) How Polymorphism is achieve in Python?

Polymorphsim is achieved thourgh multiple ways in python Duck typing, Method overriding, Function and operator overloading