**1) What is multiprocessing in python? Why is it useful?**

Ans - Multiprocessing in Python is a built-in package that allows the system to run multiple processes simultaneously. It will enable the breaking of applications into smaller threads that can run independently.

Multiprocessing is useful for CPU-bound processes, such as computationally heavy tasks since it will benefit from having multiple processors; similar to how multicore computers work faster than computers with a single core.

**2) What is the difference between multiprocessing and multithreading in python**

Ans- Multiprocessing executes many processes simultaneously, whereas multithreading executes many threads simultaneously. Multithreading uses a common address space for all the threads, whereas multiprocessing creates a separate address space for each process

**3) Write a python code to create a process using multiprocessing module?**

Ans-

import multiprocessing

def print\_cube(num):

    """

    function to print cube of given num

    """

    print("Cube: {}".format(num \* num \* num))

def print\_square(num):

    """

    function to print square of given num

    """

    print("Square: {}".format(num \* num))

if \_\_name\_\_ == "\_\_main\_\_":

    # creating processes

    p1 = multiprocessing.Process(target=print\_square, args=(10, ))

    p2 = multiprocessing.Process(target=print\_cube, args=(10, ))

    # starting process 1

    p1.start()

    # starting process 2

    p2.start()

    # wait until process 1 is finished

    p1.join()

    # wait until process 2 is finished

    p2.join()

    # both processes finished

    print("Done!")

**4) What is multiprocessing pool in python? Why is it used?**

Ans- Python multiprocessing Pool can be used for parallel execution of a function across multiple input values, distributing the input data across processes

Use the multiprocessing. Pool class when you need to be able to check on the status of tasks during their execution. Use the multiprocessing. Pool class when you need to take action based on the results of tasks, such as the first task to complete or results as they become available

5) **. How can we create a pool of worker processes in python using the multiprocessing module?**

Ans-

1)Create the Process Pool.

2)Submit Tasks to the Process Pool.

3)Wait for Tasks to Complete (Optional)

4)Shutdown the Process Pool.

**6) Write a python program to create 4 processes, each process should print a different number using the multiprocessing module in python.**

Ans- from multiprocessing import Queue

colors = ['red', 'green', 'blue', 'black']

cnt = 1

# instantiating a queue object

queue = Queue()

print('pushing items to queue:')

for color in colors:

print('item no: ', cnt, ' ', color)

queue.put(color)

cnt += 1

print('\npopping items from queue:')

cnt = 0

while not queue.empty():

print('item no: ', cnt, ' ', queue.get())

cnt += 1