PRACTICAL – 7

PROGRAM -1

AIM- Write A Python Program to Start & Join A THREAD

CODE-

|  |
| --- |
| import threading  def showA():      print("A Thread")    def showB():      print("B Thread")    t1 = threading.Thread(target=showA)  t2 = threading.Thread(target=showB)  t1.start()  t2.start()  t2.join()  print("Final Thread")  t1.join() |

OUTPUT-

|  |
| --- |
|  |

PROGRAM -2

AIM- Write A Python Program To Find Current Thread & Get ID

CODE-

|  |
| --- |
| import threading  import os  def showA():      print("A Thread")      print("Name of Current Thread is :",threading.current\_thread().name)      print("ID of FIrst Thread is :",os.getpid())    def showB():      print("B Thread")      print("Name of Second Current Thread is :",threading.current\_thread().name)      print("ID of Second Thread is :",os.getpid())    t1 = threading.Thread(target=showA,name="Thread1")  t2 = threading.Thread(target=showB,name="Thread2")  t1.start()  t2.start()  t1.join()  t2.join()  print("Final Thread") |

OUTPUT-

|  |
| --- |
|  |

PROGRAM -3

AIM- Write A Python Program To Do Mutli\_Threading Without Class

CODE-

|  |
| --- |
| from threading import \*  print(current\_thread().getName())  def mt():      print("Child Thread \n")    child=Thread(target=mt)  child.start()  print("Executing thread name :\n",current\_thread().getName()) |

OUTPUT-

|  |
| --- |
|  |

PROGRAM -4

AIM- Write A Python Program To Extend A Thread Class

CODE-

|  |
| --- |
| import threading  import time  class mythread(threading.Thread):      def run(self):          for x in range(7):              print("Hi From HarsH")    a = mythread()  a.start()  a.join()  print("Bye From",current\_thread().getName()) |

OUTPUT-

|  |
| --- |
|  |

PROGRAM -5

AIM- Write A Python Program Without Extending A Thread Class

CODE-

|  |
| --- |
| from threading import \*  class ex:      def myfunc(self):          for x in range(7):              print("CHILD")    obj=ex()  thread1=Thread(target=obj.myfunc)  thread1.start()  thread1.join()  print("DONE") |

OUTPUT:-

|  |
| --- |
|  |

PROGRAM -6

AIM- Write A Python Program With Deadlock\_with\_lock

CODE-

|  |
| --- |
| import threading  lock = threading.Lock()  def first\_function():      for i in range(5):          lock.acquire()          print('Lock Acquired')          print('Executing the First Function')          lock.release()  def second\_function():      for i in range(5):          lock.acquire()          print('Lock Acquired')          print('Executing the Second Function')          lock.release()  thread\_one = threading.Thread(target=first\_function)  thread\_two = threading.Thread(target=second\_function)  thread\_one.start()  thread\_two.start()  thread\_one.join()  thread\_two.join() |

OUTPUT-

|  |
| --- |
|  |

PROGRAM -7

AIM- Write A Python Program With Deadlock\_without\_lock

CODE-

|  |
| --- |
| import threading  def first\_function():      for i in range(5):          print('Lock Acquired')          print('Executing the First Function')    def second\_function():      for i in range(5):          print('Lock Acquired')          print('Executing the Second Function')    thread\_one = threading.Thread(target=first\_function)  thread\_two = threading.Thread(target=second\_function)  thread\_one.start()  thread\_two.start()  thread\_one.join()  thread\_two.join() |

OUTPUT:-

|  |  |
| --- | --- |
|  |  |

PROGRAM -8

AIM- Write A Python Program With Deadlock\_with\_3rd\_thread

CODE-

|  |
| --- |
| import threading  def first\_function():      for i in range(5):          print('Lock Acquired')          print('Executing the First Function')    def second\_function():      for i in range(5):          print('Lock Acquired')          print('Executing the Second Function')  thread\_one = threading.Thread(target=first\_function)  thread\_two = threading.Thread(target=second\_function)  thread\_three = threading.Thread(target=second\_function)  thread\_one.start()  thread\_two.start()  thread\_three.start()  thread\_one.join()  thread\_two.join()  thread\_three.join() |

OUTPUT-

|  |  |
| --- | --- |
|  |  |

PROGRAM - 9

AIM- Write A Python Program A Code for square in multi threading

CODE-

|  |
| --- |
| # A Code for square in multi threading  print("HARSH D ")  import threading  import time  def square(numbers):  print("calculate square numbers")  for n in numbers:  time.sleep(0.2)  print("square: ", n\*n)  def cube(numbers):  print("calculate cube numbers")  for n in numbers:  time.sleep(0.2)  print("cube: ", n\*n\*n)  arr = [2,3,8,9]  t1 = threading.Thread(target=square, args=(arr,))  t2 = threading.Thread(target=cube, args=(arr,))  t1.start()  t2.start()  t1.join()  t2.join()  print("Done") |

OUTPUT:-

|  |
| --- |
|  |

PROGRAM - 10

AIM- Write A Python Program for lock and acqiure

CODE-

|  |
| --- |
| # Write A Python Program for lock and acqiure with multi threading  import threading  import time  print("HARSH D")  class MyThread(threading.Thread):  def \_\_init\_\_(self, name, delay):  super().\_\_init\_\_()  self.name = name  self.delay = delay  def run(self):  print(f"{self.name} is running")  time.sleep(self.delay)  print(f"{self.name} is done")  # Create two threads  thread1 = MyThread("Thread 1", 2)  thread2 = MyThread("Thread 2", 1)  # Start the threads  thread1.start()  thread2.start()  # Wait for the threads to finish  thread1.join()  thread2.join()  print("All threads are done") |

OUTPUT-

|  |
| --- |
|  |