14 FEB ASS

March 5, 2023

```
[ ]: Q1. What is multithreading in python? Why is it used? Name the module used to␣
      ↪handle threads in python
```

```
[ ]: ANS -
```

```
[ ]: Multithreading allows the programmer to divide application tasks into sub-tasks␣
      ↪and simultaneously run them
     ina program. It allows threads to communicate and shares resources such as␣
      ↪files, data , and memory to the same
     processor.
```

```
[ ]: The Threading module is used to handle threads in python. the threading module␣
      ↪is a high- level implementation of
     multithreading used to deploy an application in python. To use multithreading,␣
      ↪we need to import the threading module in python
     program. A start() method is used to initiate the activity of a thread.
```

```
[ ]:
```

```
[ ]:
```

```
[ ]: Q2. Why threading module used? Write the use of the following functions
```

```
[ ]: ANS -
```

```
[ ]: The Threading module is used for Creating , Controlling and Managing threads in␣
      ↪python.
```

```
[ ]: Threading.active_count()function
      Thiss function returns the number of the Thread objects currently alive
```

```
[ ]: Threading.current_thread()
      This function will return the current Thread object,corresponding to the␣
      ↪caller thread of control.If the caller
     thread of control was not created through the threading module,then a dummy␣
      ↪thred object with limited functionality is returned.
```

[ ]:
```
Threading.enumerate()
 This method return a list of all thread objects currently alive. the list␣
 ↪includes object currently alive. the list
includes daemonic threads, dummy thread objects created by the current thread,␣
 ↪and the main thread.
```

[ ]:
```
```

[ ]:
```
```

[ ]:
```
Q3. Explain the following functions:
1. run()
2. start()
3. join()
4. isAlive()
```

[ ]:
```
ANS -
```

[ ]:
```
run():
       The.run() method executes any target function belonging to a given␣
 ↪thread object that is now active.
        It normally executes in the background after the start() methods is␣
 ↪invoked.
```

[ ]:
```
start():
        start() is where the regex was matched in str1. think of that return as␣
 ↪saying, Returning everything in str1 uo to where
    the regex was matched , and strip whitespace.
```

[ ]:
```
join():
       The join in python takes all the elements of an iterable and joins them␣
 ↪into a single string. it will return
    them into a single string. It will return the joined string. you have to␣
 ↪specify a string separator that will be used
 to seprate the concatenated string.
```

[ ]:
```
isAlive():
         is_alive()method is an inbuilt method of the thread class of the␣
 ↪threading module in python. it uses a thread object, and
        check whether that thread is alive or not, i.e, it is still running or␣
 ↪not.
```

[ ]:
```
```

[ ]:
```
```

[ ]:
```
Q4. Write a python program to create two threads. Thread one must print the␣
↪list of squares and thread
two must print the list of cubes
```

[ ]:
```
ANS -
```

[8]:
```
import threading
```

[18]:
```
import time
```

[19]:
```
def sqr(num):
    print("calcuate the square root of the given number")
    for n in num:
        time.sleep(1)
        print('square is : ',n*n)
```

[20]:
```
def cube(num):
    print("calculate the cube of the given number")
    for n in num:
        time.sleep(1)
        print("cube is : ", n*n*n)
```

[21]:
```
arr = [4,5,6,7,2]
```

[22]:
```
t1 = time.time()
```

[24]:
```
sqr(arr)
```

```
calcuate the square root of the given number
square is :  16
square is :  25
square is :  36
square is :  49
square is :  4
```

[25]:
```
cube(arr)
```

```
calculate the cube of the given number
cube is :  64
cube is :  125
cube is :  216
cube is :  343
cube is :  8
```

[ ]:
```
Q5. State advantages and disadvantages of multithreading
```

```
[ ]: ANS -
```

```
[ ]: Advantages:
         Multithreading in python has serval adavntages , making it a popular␣
       ↪approach.
         - Python multithreading enables efficient utilization of the resources as␣
       ↪the threads share the data space and memory
         - Multithreading in python allows the concurrent and parallel occurance of␣
       ↪various tasks.
         - It causes a reduction in time consumption or response time , there by␣
       ↪increasing the performance.
```

```
[ ]: Disadvantages :
         - Difficulty of writing code. Multithreaded and multicontexted applications␣
       ↪are not easy to write.
         - Difficulty of debugging , it is much harder to replicate an error in a␣
       ↪multithreaded application than it is to do so in a single
           threaded.
         - Difficulty of testing. testing a multithreaded application is more␣
       ↪difficult than testing a single threaded application .
         - Difficulty of managing concurrency. the task of managing concurrency␣
       ↪among threads is difficult and has the potential to introduce
           new problems in to application.
```

```
[ ]:
```

```
[ ]:
```

```
[ ]: Q6. Explain deadlocks and race conditions.
```

```
[ ]: ANS -
```

```
[ ]: race conditions:
                         A race condition means that your program result might depend on␣
       ↪the oder of execution of the individual steps.
             this becomes essential in concurrent progams that can run multiple␣
       ↪threads of execution simultaneously through context switching.
             and they access a shared resource like variable.
```

```
[ ]: dead lock condition:
                             When two processes are waiting for each other directly or␣
       ↪indirectly , it is called deadlock.
             This usually occurs when two processes are waiting for shared resorces␣
       ↪acquired by others.A dead lock is a concurrency failure mode
             where a thread or thread wait for a condition that never occurs. the␣
       ↪result is the deadlock threads are unable to progress and the
```

```
        program is stuck ot frozen and must be terminated forcefully.
```

[ ]:

```

```