CH1

Assessment on Multithreaded Programming in Python

1. What is a thread in the context of programming?

a. A thread is a type of computer processor

b. A thread is an individual and separate unit of execution that is part of a process

c. A thread is a large block of memory allocated for a program

d. A thread is always a standalone program

2. Why is threading important in programming?

a. Threading helps reduce the size of the code

b. Threading allows multiple programs to run simultaneously on a computer

c. Threading enables tasks to be handled in parallel and keeps the program responsive

d. Threading has no significant impact on program performance

3. What is the key difference between a thread and a process?

a. Threads share the same context while processes have isolated contexts

b. Processes are used for graphics tasks while threads are used for networking tasks

c. Processes are lightweight entities while threads are heavy entities

d. Threads run on multiple CPUs while processes run on a single CPU

4. Which module in Python provides support for threads?

a. Multi-threading module

b. Threadable module

c. Threading module

d. Multi-process module

5. What is the purpose of the join() method in Python threading?

a. To start a new thread

b. To terminate a thread

c. To wait for threads to terminate

d. To create a shared resource for threads

6. In Python, how can a new thread be created using the threading module?

a. Invoking the sleep() method

b. Creating an instance of the Thread class and calling the start() method

c. Utilizing the _thread module

d. Using the join() method

7. What does the Lock function in Python threading do?

a. Exits the thread

b. Toggles between locked and unlocked states

c. Allocates a LockType object for synchronization
d. Sets a delay in thread execution

8. Which statement best describes the Global Interpreter Lock (GIL) in Python?

a. GIL increases the speed of programs by allowing multiple threads to run simultaneously
b. GIL ensures that only one thread executes in the interpreter at a time
c. GIL is exclusively used for memory management in Python
d. GIL releases the locks on shared resources for threads

9. When creating threads in Python, what is the purpose of acquiring and releasing locks?

a. To synchronize thread access to shared resources
b. To prevent threading errors
c. To terminate the program
d. To allocate memory resources

10. What is the advantage of using threading.Lock() in Python?

a. It allows threads to run without synchronization
b. It introduces delays in thread execution
c. It creates a shared resource for all threads
d. It synchronizes thread access to shared resources

11. What does the threading.Thread(target=function, args=args, kwargs=kwargs).start() function do in Python?

a. Exits the thread
b. Spawns a new thread and executes a function with given arguments
c. Delays the thread execution
d. Determines the thread priority

12. Why should the threading module be preferred over the thread module for thread creation?

a. The threading module provides low-level thread control
b. The thread module is more efficient for large-scale programs
c. The threading module provides high-level support for threads
d. The thread module has less overhead in performance

Answer Key (Always review AI generated answers for accuracy - Math is more likely to be inaccurate)

1. b
2. c
3. a

CH1

4. c
5. c
6. b
7. c
8. b
9. a
10. d
11. b
12. c