1. \*\*Which class in C# is used to create and manage threads?\*\*

- A) `ThreadManager`

- B) `Thread`

- C) `Task`

- D) `ThreadPool`

\*\*Answer:\*\* B) `Thread`

2. \*\*What method is used to start a thread in C#?\*\*

- A) `Begin()`

- B) `Run()`

- C) `Start()`

- D) `Execute()`

\*\*Answer:\*\* C) `Start()`

3. \*\*What is the default priority of a thread in C#?\*\*

- A) `Low`

- B) `Normal`

- C) `High`

- D) `AboveNormal`

\*\*Answer:\*\* B) `Normal`

4. \*\*Which class provides a way to synchronize access to a shared resource in C#?\*\*

- A) `Thread`

- B) `Monitor`

- C) `Task`

- D) `Mutex`

\*\*Answer:\*\* B) `Monitor`

5. \*\*Which method in the `Monitor` class is used to acquire a lock on an object?\*\*

- A) `Enter()`

- B) `Lock()`

- C) `Acquire()`

- D) `Begin()`

\*\*Answer:\*\* A) `Enter()`

6. \*\*What is the purpose of the `Mutex` class in multi-threading?\*\*

- A) To provide thread-safe collections

- B) To lock a block of code for a single thread

- C) To synchronize access across multiple processes

- D) To handle exceptions in threads

\*\*Answer:\*\* C) To synchronize access across multiple processes

7. \*\*Which statement is true about `ThreadPool`?\*\*

- A) It creates new threads for each task.

- B) It reuses a pool of threads for executing tasks.

- C) It provides direct control over thread execution.

- D) It only works with `Task` objects.

\*\*Answer:\*\* B) It reuses a pool of threads for executing tasks.

8. \*\*What does the `Join()` method do in the `Thread` class?\*\*

- A) Starts the thread

- B) Suspends the current thread until the target thread completes

- C) Cancels the target thread

- D) Sets the priority of the thread

\*\*Answer:\*\* B) Suspends the current thread until the target thread completes

9. \*\*Which of the following is a type of thread-safe collection in C#?\*\*

- A) `List<T>`

- B) `Dictionary<TKey, TValue>`

- C) `ConcurrentQueue<T>`

- D) `HashSet<T>`

\*\*Answer:\*\* C) `ConcurrentQueue<T>`

10. \*\*What is the primary purpose of `async` and `await` keywords in C#?\*\*

- A) To create new threads

- B) To simplify asynchronous programming

- C) To handle thread synchronization

- D) To monitor thread performance

\*\*Answer:\*\* B) To simplify asynchronous programming

11. \*\*Which class is used to manage a thread pool in C#?\*\*

- A) `ThreadPool`

- B) `Task`

- C) `Thread`

- D) `ThreadManager`

\*\*Answer:\*\* A) `ThreadPool`

12. \*\*Which method is used to queue a task to the thread pool in C#?\*\*

- A) `QueueTask()`

- B) `Run()`

- C) `QueueUserWorkItem()`

- D) `Submit()`

\*\*Answer:\*\* C) `QueueUserWorkItem()`

13. \*\*What does the `Volatile` keyword indicate in C#?\*\*

- A) The variable is thread-safe

- B) The variable's value might be changed by different threads

- C) The variable is read-only

- D) The variable is synchronized

\*\*Answer:\*\* B) The variable's value might be changed by different threads

14. \*\*What is the purpose of the `Lock` statement in C#?\*\*

- A) To pause the execution of the thread

- B) To synchronize access to a block of code

- C) To start a new thread

- D) To terminate a thread

\*\*Answer:\*\* B) To synchronize access to a block of code

15. \*\*Which of the following is NOT a thread synchronization primitive in C#?\*\*

- A) `Semaphore`

- B) `EventWaitHandle`

- C) `SpinLock`

- D) `ThreadLocal`

\*\*Answer:\*\* D) `ThreadLocal`

16. \*\*What is the main difference between `Task` and `Thread` in C#?

16. \*\*What is the main difference between `Task` and `Thread` in C#?\*\*

- A) `Task` is used for CPU-bound work, while `Thread` is used for I/O-bound work.

- B) `Task` represents an asynchronous operation and provides more control over scheduling, while `Thread` represents a unit of execution.

- C) `Task` is a low-level abstraction, while `Thread` is a high-level abstraction.

- D) `Task` can only be used in async methods, while `Thread` can be used anywhere.

\*\*Answer:\*\* B) `Task` represents an asynchronous operation and provides more control over scheduling, while `Thread` represents a unit of execution.

17. \*\*Which method is used to cancel a `Task` in C#?\*\*

- A) `Cancel()`

- B) `Stop()`

- C) `Dispose()`

- D) `CancellationToken.Cancel()`

\*\*Answer:\*\* D) `CancellationToken.Cancel()`

18. \*\*What is the primary benefit of using `Parallel.For` and `Parallel.ForEach` over traditional looping constructs?\*\*

- A) They provide thread-local storage.

- B) They automatically handle synchronization.

- C) They automatically parallelize the loop and can improve performance on multi-core processors.

- D) They support synchronous operations.

\*\*Answer:\*\* C) They automatically parallelize the loop and can improve performance on multi-core processors.

19. \*\*Which of the following methods is used to pause the execution of a thread for a specific time period?\*\*

- A) `Sleep()`

- B) `Pause()`

- C) `Delay()`

- D) `Wait()`

\*\*Answer:\*\* A) `Sleep()`

20. \*\*What does the `ThreadStatic` attribute do in C#?\*\*

- A) It marks a field to be thread-specific, meaning each thread will have its own instance of the field.

- B) It makes a field immutable across threads.

- C) It makes a field accessible only to the main thread.

- D) It ensures a field is synchronized across all threads.

\*\*Answer:\*\* A) It marks a field to be thread-specific, meaning each thread will have its own instance of the field.