

Homework

The following homework is designed to cover the course objectives for this unit.

Assignment 3.1:

Research the ITT Tech Virtual Library to find out what the Thread.isAlive() method tells you about a particular thread. Present your findings in the form of a report listing the features of the Thread.isAlive() method. Cite your sources using Chicago Manual of Style format. Submit your report to your instructor at the beginning of Unit 4.

Assignment 3.2:

Research the ITT Tech Virtual Library to find out the difference between a daemon thread and a user thread. Cite your sources using Chicago Manual of Style format. Submit your report to your instructor at the beginning of Unit 4.

Assignment 3.3:

Research the ITT Tech Virtual Library to find out how to create a new thread without specifying its group in the constructor. Cite your sources using Chicago Manual of Style format. Submit your report to your instructor at the beginning of Unit 4.

Assignment 3.4:

Research the ITT Tech Virtual Library to find out the state of a thread when it blocks an I/O and the initial state of a thread after it is created and started. Cite your sources using Chicago Manual of Style format. Submit your report to your instructor at the beginning of Unit 4.

Assignment 3.5:

Research the ITT Tech Virtual Library to find out if it is possible to call the wait() method in a non-synchronized block. Cite your sources using Chicago Manual of Style format. Submit your report to your instructor at the beginning of Unit 4.

Assignment 3.6:

Answer the following questions and submit them to your instructor at the beginning of Unit 4. **Note:** Some questions may go across multiple pages; be sure to read the entire question and all answer options.

1. You can use the _____ method to temporarily release time for other threads.
(Hint: There are multiple answers.)
 - a. suspend()
 - b. yield()
 - c. stop()
 - d. sleep(long milliseconds)
2. Which of the following statements are true? (Hint: There are multiple answers.)
 - a. A blocking queue has a capacity.
 - b. A blocking queue causes a thread to block when you try to add an element to a full queue.
 - c. A blocking queue causes a thread to block when you try to remove an element from an empty queue.
 - d. The BlockingQueue interface is the base interface for all concrete blocking queue classes.
 - e. The BlockingQueue interface provides the synchronized put and take methods for adding an element to the head of the queue and for removing an element from the tail of the queue.
3. Suppose there are three Runnable tasks: task1, task2, task3. How do you run them in a thread pool with two fixed threads?
 - a. `ExecutorService executor = Executors.newFixedThreadPool(3); executor.execute(task1); executor.execute(task2); executor.execute(task3);`
 - b. `new Thread(task1).start(); new Thread(task2).start(); new Thread(task3).start();`
 - c. `ExecutorService executor = Executors.newFixedThreadPool(1); executor.execute(task1); executor.execute(task2); executor.execute(task3);`
 - d. `ExecutorService executor = Executors.newFixedThreadPool(2); executor.execute(task1); executor.execute(task2); executor.execute(task3);`
4. Analyze the following code:

```
public class Test implements Runnable {
    public static void main(String[] args) {
        Test t = new Test();
    }

    public Test() {
        Thread t = new Thread(this);
        t.start();
    }

    public void run() {
```

```

        System.out.println("test");
    }
}

```

Which of the following sentences is true?

- a. The program will have a compilation error because t is defined in both the main() method and the constructor Test().
 - b. The program will compile fine, but it will not run because you cannot use the keyword this in the constructor.
 - c. The program will compile and run and it will display test.
 - d. The program will compile and run and it will display nothing.
5. Given the following code, which of the following sets of code can you use to replace the comment so that the program displays time to the console every second?

```

import java.applet.*;
import java.util.*;

public class Test extends Applet implements Runnable {
    public void init() {
        Thread t = new Thread(this);
        t.start();
    }

    public void run() {
        for(;;) {
            //display time every second
            System.out.println(new Date().toString());
        }
    }
}

```

- a. `try { sleep(1000); } catch(InterruptedException e) { }`
- b. `try { Thread.sleep(1000); } catch(InterruptedException e) { }`
- c. `try { Thread.sleep(1000); } catch(RuntimeException e) { }`
- d. `try { t.sleep(1000); } catch(InterruptedException e) { }`

6. Analyze the following code:

```

public class Test implements Runnable {
    public static void main(String[] args) {
        Test t = new Test();
        t.start();
    }

    public void run() {
    }
}

```

Which of the following sentences is true?

- a. The program will not compile because the start() method is not defined in the Test class.
 - b. The program will compile, but it will not run because the start() method is not defined.
 - c. The program will compile, but it will not run because the run() method is not implemented.
 - d. The program will compile and run fine.
7. Which method on a condition should you invoke to wake all waiting threads?
- a. condition.wakeAll();
 - b. condition.wake();
 - c. condition.signal();
 - d. condition.signalAll();
8. How do you create a cached thread pool?
- a. `ExecutorService executor = Executors.newCachedThreadPool(1);`
 - b. `ExecutorService executor = Executors.newCachedThreadPool();`
 - c. `ExecutorService executor = Executors.newCachedThreadPool(3);`
 - d. `ExecutorService executor = Executors.newCachedThreadPool(2);`
9. Which of the following statements are defined in the Object class?
(Hint: There are multiple answers.)
- a. notify()
 - b. wait()
 - c. toString()
 - d. sleep(long milliseconds)
 - e. notifyAll()
10. You can use the _____ method to force one thread to wait for another thread to finish.
- a. join()
 - b. stop()
 - c. yield()
 - d. suspend()
 - e. sleep(long milliseconds)

11. Which of the following statements are true? (Hint: There are multiple answers.)
- a. A synchronized statement is placed inside a synchronized block.
 - b. A synchronized instance method acquires a lock on the class of the object for which the method was invoked.
 - c. A synchronized instance method acquires a lock on the object for which the method was invoked.
 - d. A synchronized statement can be used to acquire a lock on any object, not just this object, when executing a block of the code in a method.
12. You can create a blocking queue using _____.
(Hint: There are multiple answers.)
- a. ArrayBlockingQueue
 - b. LinkedBlockingQueue
 - c. PriorityBlockingQueue
 - d. PriorityQueue
13. Which of the following statements are true? (Hint: There are multiple answers.)
- a. A condition is associated with a lock.
 - b. The signal method on a condition causes the lock for the condition to be released.
 - c. Once you invoke the await method on a condition, the lock is automatically released. Once the condition is right, the thread re-acquires the lock and continues executing.
 - d. To invoke methods on a condition, the lock must be obtained first.
14. Which of the following statements are true? (Hint: There are multiple answers.)
- a. Semaphores can be used to restrict the number of threads that access a shared resource.
 - b. After finishing with the resource, the thread must return the permit back to the semaphore.
 - c. Before accessing the resource, a thread must acquire a permit from the semaphore.
 - d. You can create a Semaphore with a specified number of permits.
15. What is the output of the following code?

```
// Test.java: Define threads using the Thread class
public class Test {
    /** Main method */
    public static void main(String[] args) {
        new Test();
    }

    public Test() {
        // Create threads
    }
}
```

```

PrintChar printA = new PrintChar('a', 4);
PrintChar printB = new PrintChar('b', 4);

// Start threads
printA.run();
printB.run();
}

class PrintChar implements Runnable {
    private char charToPrint; // The character to print
    private int times; // The times to repeat

    /** Construct a thread with specified character and number of
        times to print the character
    */
    public PrintChar(char c, int t) {
        charToPrint = c;
        times = t;
    }

    /** Override the run() method to tell the system
        what the thread will do
    */
    public void run() {
        for (int i = 0; i < times; i++)
            System.out.print(charToPrint);
    }
}

```

- a. aaaaabbbbb
- b. bbbbaaaaa
- c. ababababab
- d. bababababa

16. Which of the following statements are true? (Hint: There are multiple answers.)

- a. You can use a timer or a thread to control animation.
- b. In general, threads are more reliable and responsive than timers.
- c. The timer and event-handling run on the same event dispatcher thread. If it takes a long time to handle the event, the actual delay time between two events will be longer than the requested delay time.
- d. A timer is a source component that fires an `ActionEvent` at a “fixed rate.”

17. Which method on a condition should you invoke to causes the current thread to wait until the condition is signaled?

- a. `condition.wait();`
- b. `condition.waited();`
- c. `condition.waiting();`
- d. `condition.await();`

18. When you run the following program, what will happen?

```
public class Test extends Thread {  
    public static void main(String[] args) {  
        Test t = new Test();  
        t.start();  
        t.start();  
    }  
  
    public void run() {  
        System.out.println("test");  
    }  
}
```

- a. The program will display test once.
 - b. The program will display test twice.
 - c. Nothing will be displayed.
 - d. An illegal java.lang.IllegalThreadStateException may be thrown because you just started thread and thread might have not yet finished before you start it again.
19. Which of the following statements are true? (Hint: There are multiple answers.)
- a. The javax.swing.SwingUtilities.invokeLaterAndWait method runs the code in the event dispatcher thread.
 - b. The javax.swing.SwingUtilities.invokeLater method creates a thread.
 - c. The javax.swing.SwingUtilities.invokeLater method runs the code in the event dispatcher thread and doesn't return until the event-dispatching thread has executed the specified code.
 - d. GUI event handling is executed in the event dispatcher thread.
20. Which of the following methods in Thread throws InterruptedException?
- a. run()
 - b. sleep(long)
 - c. yield()
 - d. start()
 - e. setPriority(int)