**Question 1**

class A extends Thread {

private int i;

public void run() {i = 1;}

public static void main(String[] args) {

A a = new A(); a.start(); System.out.print(a.i);

}}

What are the possible results of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints nothing |
| b. | Prints: 0 |
| c. | Prints: 1 |
| d. | Prints: 01 |
| e. | Prints: 10 |
| f. | Compile-time error |
| g. | Run-time error |

**Question 2**

class A extends Thread {

private int i;

public void run() {i = 1;}

public static void main(String[] args) {

A a = new A(); a.run(); System.out.print(a.i);

}}

What is the result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints nothing |
| b. | Prints: 0 |
| c. | Prints: 1 |
| d. | Prints: 01 |
| e. | Prints: 10 |
| f. | Compile-time error |
| g. | Run-time error |
| h. | None of the above |

**Question 3**

class A extends Thread {

public void run() {

try {sleep(10000);} catch (InterruptedException ie){}

}

public static void main(String[] args) {

A a1 = new A();

long startTime = System.currentTimeMillis();

a1.start();

System.out.print(System.currentTimeMillis() - startTime);

}}

What are the possible results of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints a number greater than or equal to 0 |
| b. | The number printed must always be greater than 10000 |
| c. | This program will run for at least ten seconds |
| d. | Compile-time error |
| e. | Run-time error |

**Question 4**

Which of the following is used to force each thread to reconcile its working copy of a variable with the master copy in main memory?

|  |  |
| --- | --- |
| a. | final |
| b. | static |
| c. | synchronized |
| d. | transient |
| e. | volatile |
| f. | native |

**Question 5**

class A extends Thread {

public void run() {

synchronized (this) {

try {wait(5000);} catch (InterruptedException ie){}

}}

public static void main(String[] args) {

A a1 = new A();

long startTime = System.currentTimeMillis();

a1.start();

System.out.print(System.currentTimeMillis() - startTime + ",");

try {a1.join(6000);} catch (InterruptedException ie) {}

System.out.print(System.currentTimeMillis() - startTime);

}}

What are the possible results of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | The first number printed is greater than or equal to 0 |
| b. | The first number printed must always be greater than 5000 |
| c. | The second number printed must always be greater than 5000 |
| d. | The second number printed must always be greater than 6000 |
| e. | The synchronized block inside the run method is not necessary |
| f. | Compile-time error |
| g. | Run-time error |

**Question 6**

class A extends Thread {

String[] sa;

public A(String[] sa) {this.sa = sa;}

public void run() {

synchronized (sa) {System.out.print(sa[0] + sa[1] + sa[2]);}

}}

class B {

private static String[] sa = new String[]{"X","Y","Z"};

public static void main (String[] args) {

synchronized (sa) {

Thread t1 = new A(sa); t1.start();

sa[0] = "A"; sa[1] = "B"; sa[2] = "C";

}}}

What is the result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints: XYZ |
| b. | Prints: AYZ |
| c. | Prints: ABZ |
| d. | Prints: ABC |
| e. | Compile-time error |
| f. | Run-time error |
| g. | None of the above |

**Question 7**

class A extends Thread {

String[] sa;

public A(String[] sa) {this.sa = sa;}

public void run() {

synchronized (sa) {

while (!sa[0].equals("Done")) {

try {sa.wait();} catch (InterruptedException ie) {}

}}

System.out.print(sa[1] + sa[2] + sa[3]);

}}

class B {

private static String[] sa = new String[]{"Not Done","X","Y","Z"};

public static void main (String[] args) {

Thread t1 = new A(sa); t1.start();

synchronized (sa) {

sa[0] = "Done";

sa[1] = "A"; sa[2] = "B"; sa[3] = "C";

sa.notify();

}}}

What is the result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints: XYZ |
| b. | Prints: AYZ |
| c. | Prints: ABZ |
| d. | Prints: ABC |
| e. | Compile-time error |
| f. | Run-time error |
| g. | None of the above |

**Question 8**

Which of the following are true statements?

|  |  |
| --- | --- |
| a. | The Thread.join method is static |
| b. | The Thread.join method is always invoked on an instance of Thread |
| c. | The Thread.join method causes the current thread to wait for the referenced thread to die |
| d. | The Thread.join method declares an InterruptedException in the throws clause |
| e. | The Thread.join method accepts a timeout value as an argument |
| f. | The timeout value sets the minimum time that the current thread will wait for the death of the referenced thread |
| g. | Thread.join will return immediately if the timeout value is zero |
| h. | A timeout of zero will allow Thread.join to wait forever if necessary |

**Question 9**

Which of the following allows a thread t1 to become the holder of the lock of object obj1.

|  |  |
| --- | --- |
| a. | By blocking on I/O |
| b. | By entering a synchronized instance method of the obj1 |
| c. | By invoking the wait method on the object |
| d. | By entering the body of a block that is synchronized on obj1 |
| e. | By entering a synchronized static method of the obj1 |
| f. | By invoking the notify method on obj1 |

**Question 10**

After invoking the wait method on an object, obj1, a thread, T1, will remain in the wait set of obj1 until which of the following occurs?

|  |  |
| --- | --- |
| a. | Another thread invokes the notify method on the object, obj1, and T1 is selected to move out of the wait set |
| b. | Another thread invokes the notifyAll method on the object |
| c. | Another thread invokes the resume method on thread T1 |
| d. | Another thread interrupts thread T1 |
| e. | The priority of thread T1 is increased |
| f. | A specified timeout period has elapsed |
| g. | Another thread invokes the join method on thread T1 |

**Question 11**

class A implements Runnable{public void run() {}}

class B {

public static void main(String[] args) {

Thread t1 = new Thread(); // 1

Thread t2 = new Thread(new A()); // 2

Thread t3 = new Thread(new A(), "A"); // 3

Thread t4 = new Thread("A"); // 4

}}

A compile-time error is generated at which line?

|  |  |
| --- | --- |
| a. | 1 |
| b. | 2 |
| c. | 3 |
| d. | 4 |
| e. | None of the above |

**Question 12**

class A implements Runnable{public void run() {}}

class B {

public static void main(String[] args) {

Thread t1 = new Thread(); // 1

Thread t2 = new Thread(new A()); // 2

Thread t3 = new Thread("A", new A()); // 3

Thread t4 = new Thread("A"); // 4

}}

A compile-time error is generated at which line?

|  |  |
| --- | --- |
| a. | 1 |
| b. | 2 |
| c. | 3 |
| d. | 4 |
| e. | None of the above |

**Question 13**

class A extends Thread {

private boolean done;

public void setDone(boolean done) {this.done = done;}

public void run() {

synchronized (this) {

while (!done) {try {wait();} catch (InterruptedException ie){}}

}}

public static void main(String[] args) {

A a1 = new A();

long startTime = System.currentTimeMillis();

a1.start();

System.out.print(System.currentTimeMillis() - startTime);

}}

Which is a possible result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | The number printed is greater than or equal to 0 |
| b. | The synchronized block inside the run method is not necessary |
| c. | This program runs to completion after the elapsed time is printed |
| d. | Compile-time error |
| e. | Run-time error |
| f. | None of the above |

**Question 14**

class A extends Thread {

public void run() {

synchronized (this) {

try {wait();} catch (InterruptedException ie){}

}}

public static void main(String[] args) {

A a1 = new A(); a1.setDaemon(true);

long startTime = System.currentTimeMillis();

a1.start();

System.out.print(System.currentTimeMillis() - startTime + ",");

}}

Which is a possible result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | The number printed is greater than or equal to 0 |
| b. | The synchronized block inside the run method is not necessary |
| c. | Thread a1 waits forever and the program runs forever |
| d. | Compile-time error |
| e. | Run-time error |
| f. | None of the above |

**Question 15**

class A extends Thread {

private Object obj;

public A(Object obj) {this.obj = obj;}

public void run() {

try {

synchronized (obj) {obj.wait();}

} catch (InterruptedException ie) {}

System.out.print(Thread.currentThread().getName());

}}

class B {

private void m1() {

for (int i = 0; i < 10; i++) {

A t1 = new A(this);

t1.setName(String.valueOf(i)); t1.setDaemon(true); t1.start();

}

synchronized (this) {notifyAll();}

}

public static void main(String[] args) {new B().m1();}

}

What are the possible results of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | All of the numbers 0 through 9 must always be printed |
| b. | Some or all of the numbers 0 through 9 could be printed |
| c. | Nothing is printed |
| d. | Run-time error |

**Question 16**

class C extends Thread {

private static String[] sa = new String[]{"Not Done","X","Y","Z"};

public void run() {

synchronized (sa) {

while (!sa[0].equals("Done")) {

try {sa.wait();} catch (InterruptedException ie) {}

}}

System.out.print(sa[1] + sa[2] + sa[3]);

}

public static void main (String[] args) {

start();

synchronized (sa) {

sa[0] = "Done";

sa[1] = "A"; sa[2] = "B"; sa[3] = "C";

sa.notify();

}}}

Which is a possible result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints: XYZ |
| b. | Prints: AYZ |
| c. | Prints: ABZ |
| d. | Prints: ABC |
| e. | Compile-time error |
| f. | Run-time error |
| g. | None of the above |

**Question 17**

class C extends Thread {

private static String[] sa = new String[]{"Not Done","X","Y","Z"};

public void run() {

synchronized (this) {

while (!sa[0].equals("Done")) {

try {wait();} catch (InterruptedException ie) {}

}}

System.out.print(sa[1] + sa[2] + sa[3]);

}

void m1() {

start();

synchronized (this) {

sa[0] = "Done";

sa[1] = "A"; sa[2] = "B"; sa[3] = "C";

}}

public static void main (String[] args) {

new C().m1(); notify();

}}

Which is a possible result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints: XYZ |
| b. | Prints: AYZ |
| c. | Prints: ABZ |
| d. | Prints: ABC |
| e. | Compile-time error |
| f. | Run-time error |
| g. | None of the above |

**Question 18**

class A extends Thread {

public void run() {

long startTime = System.currentTimeMillis();

long endTime = startTime + 10000;

while (System.currentTimeMillis() < endTime) {

yield();

}}

public static void main(String[] args) {

A a1 = new A();

long startTime = System.currentTimeMillis();

a1.start(); sleep(1000); a1.interrupt(); a1.join();

System.out.print(System.currentTimeMillis() - startTime);

}}

Which is a possible result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints a number that is less than 1000 |
| b. | Prints a number between 1000 and 9999 |
| c. | Prints a number larger than 10000 |
| d. | Compile-time error |
| e. | Run-time error |
| f. | None of the above |

**Question 19**

class A extends Thread {

public void run() {System.out.print("A");}

}

class B {

public static void main (String[] args) {

A a = new A(); a.start();

try {

a.join();

} catch (InterruptedException ie) {ie.printStackTrace();}

a.start(); // 1

}}

What is the result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | The program compiles and runs without error |
| b. | The second attempt to start thread t1 is successful |
| c. | The second attempt to start thread t1 is ignored |
| d. | Compile-time error at marker 1 |
| e. | An IllegalThreadStateException is thrown at run-time |
| f. | None of the above |

**Question 20**

class A extends Thread {

private static B b = new B();

private String s1;

public void run() {System.out.print(b.m1(s1));}

A(String threadName, String s1) {

super(threadName); this.s1 = s1;

}

public static void main (String[] args) {

A a = new A("T1","A"), b = new A("T2","B"); a.start(); b.start();

}}

class B {

private String s1;

public synchronized String m1(String s) {

s1 = s;

try {Thread.sleep(1);} catch (InterruptedException ie) {}

return "["+Thread.currentThread().getName()+","+s1+"]";

}}

What are the possible results of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints nothing |
| b. | Prints: [T1,A][T2,B] |
| c. | Prints: [T1,B][T2,B] |
| d. | Prints: [T2,B][T1,A] |
| e. | Prints: [T2,A][T1,A] |
| f. | Compile-time error |
| g. | Run-time error |

**Question 21**

class A extends Thread {

static long startTime;

public void run() {

for (int i = 0; i < 99999; i++) {Math.sin(i);}

String name = Thread.currentThread().getName();

long time = System.currentTimeMillis();

System.out.println(name + " done at " + (time - startTime));

}

public static void main(String[] args) {

A t1 = new A(); A t2 = new A();

t1.setName("T1"); t2.setName("T2");

t1.setPriority(Thread.MIN\_PRIORITY);

t2.setPriority(Thread.MAX\_PRIORITY);

startTime = System.currentTimeMillis();

t1.start(); t2.start();

}}

Which of the following is a true statement?

|  |  |
| --- | --- |
| a. | The priority assigned to thread T2 is greater than the priority assigned to T1 |
| b. | Java guarantees that thread T2 will get more CPU time than T1 |
| c. | Java guarantess that thread T2 will run to completion before T1 |
| d. | None of the above |