JAVA means DURGA SOFT

SCJP / OCJP

Question Bank

Chapter 9 : Multi-Threading

DURGA M.Tech

(Sun certified & Realtime Expert)

Ex. IBM Employee

Trained Lakhs of Students
for last 14 years across INDIA

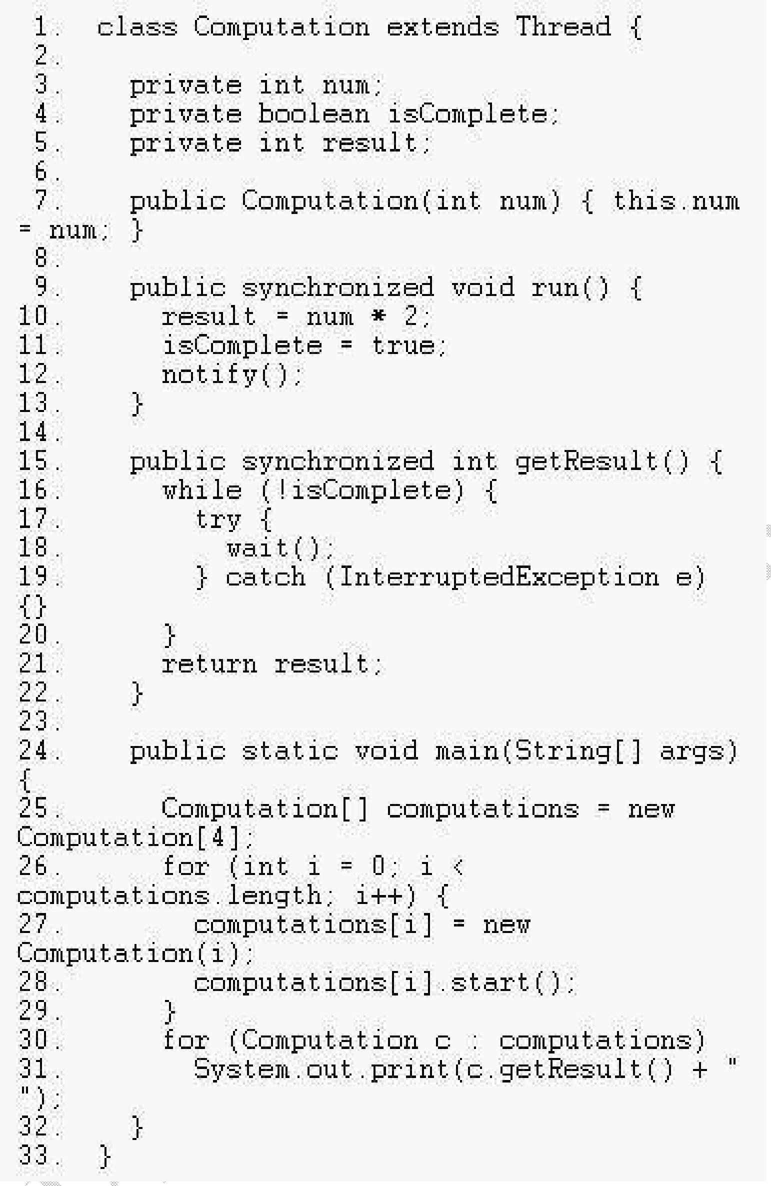
India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

9. Multi-Threading

Q: 01 Click the Exhibit button.
What is the result?



```
class Computation extends Thread {

  private int num;
  private boolean isComplete;
  private int result;

  public Computation(int num) { this.num = num; }

  public synchronized void run() {
    result = num * 2;
    isComplete = true;
    notify();
  }

  public synchronized int getResult() {
    while (!isComplete) {
      try {
        wait();
      } catch (InterruptedException e) {}
    }
    return result;
  }

  public static void main(String[] args) {
    Computation[] computations = new Computation[4];
    for (int i = 0; i < computations.length; i++) {
      computations[i] = new Computation(i);
      computations[i].start();
    }
    for (Computation c : computations)
      System.out.print(c.getResult() + " ");
  }
}
```

A. The code will deadlock.
B. The code may run with no output.
C. An exception is thrown at runtime.
D. The code may run with output "0 6".
E. The code may run with output "2 0 6 4".
F. The code may run with output "0 2 4 6".

Answer: F

Q: 02 Given:

```
public class Threads2 implements Runnable {

public void run() {
System.out.println("run.");
throw new RuntimeException("Problem");
}
public static void main(String[] args) {
Thread t = new Thread(new Threads2());
t.start();
System.out.println("End of method.");
}
}
```

Which two can be results? (Choose two.)

A. java.lang.RuntimeException: Problem

B. run.
java.lang.RuntimeException: Problem

C. End of method.
java.lang.RuntimeException: Problem

D. End of method.
run.
java.lang.RuntimeException: Problem

E. run.
java.lang.RuntimeException: Problem
End of method.

Answer: D, E

Q: 03 Given:

```
public class TestSeven extends Thread {
private static int x;
public synchronized void doThings() {
int current = x;
current++;
x = current;
}
public void run() {
doThings();
}
}
```

Which statement is true?

A. Compilation fails.

B. An exception is thrown at runtime.

C. Synchronizing the run() method would make the class thread-safe.

D. The data in variable "x" are protected from concurrent access problems.

E. Declaring the doThings() method as static would make the class thread-safe.

F. Wrapping the statements within doThings() in a synchronized(new Object()) { } block would make the class
thread-safe.

Answer: E

Q: 04 Given:

```
public class Threads3 implements Runnable {
public void run() {
System.out.print("running");
}
public static void main(String[] args) {
Thread t = new Thread(new Threads3());
t.run();
t.run();
t.start();
}
}
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. The code executes and prints "running".
D. The code executes and prints "runningrunning".
E. The code executes and prints "runningrunningrunning".

Answer: E

Q: 05 Given:

```
public class NamedCounter {
private final String name;
private int count;
public NamedCounter(String name) { this.name = name; }
public String getName() { return name; }
public void increment() { count++; }
public int getCount() { return count; }
public void reset() { count = 0; }
}
```

Which three changes should be made to adapt this class to be used safely by multiple threads? (Choose three.)

A. declare reset() using the synchronized keyword
B. declare getName() using the synchronized keyword
C. declare getCount() using the synchronized keyword
D. declare the constructor using the synchronized keyword
E. declare increment() using the synchronized keyword

Answer: A, C, E

Q: 06 Given:

```
7. void waitForSignal() {
8. Object obj = new Object();
9. synchronized (Thread.currentThread()) {
10. obj.wait();
11. obj.notify();
12. }
13. }
```

Which statement is true?

A. This code may throw an InterruptedException.
B. This code may throw an IllegalStateException.
C. This code may throw a TimeoutException after ten minutes.
D. This code will not compile unless "obj.wait()" is replaced with "((Thread) obj).wait()".
E. Reversing the order of obj.wait() and obj.notify() may cause this method to complete normally.
F. A call to notify()or notifyAll() from another thread may cause this method to complete normally.

Answer: B

Q: 07 Which two code fragments will execute the method doStuff() in a separate thread? (Choose two.)

```
A. new Thread() {
public void run() { doStuff(); }
};
B. new Thread() {
public void start() { doStuff(); }
};
C. new Thread() {
public void start() { doStuff(); }
}.run();
D. new Thread() {
public void run() { doStuff(); }
}.start();
E. new Thread(new Runnable() {
public void run() { doStuff(); }
}).run();
F. new Thread(new Runnable() {
public void run() { doStuff(); }
}).start();
```

Answer: D, F

Q: 08 Given:

```
1. public class TestOne implements Runnable {
2. public static void main (String[] args) throws Exception {
3. Thread t = new Thread(new TestOne());
4. t.start();
5. System.out.print("Started");
6. t.join();
7. System.out.print("Complete");
8. }
9. public void run() {
10. for (int i = 0; i < 4; i++) {
```

```
System.out.print(i);
}
}
}
```

What can be a result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes and prints "StartedComplete".

D. The code executes and prints "StartedComplete0123".

E. The code executes and prints "Started0123Complete".

Answer: E

Q: 09Given:

```
public class TestOne {
public static void main (String[] args) throws Exception {
Thread.sleep(3000);
System.out.println("sleep");
}
}
```

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes normally and prints "sleep".

D. The code executes normally, but nothing is printed.

Answer: C

Q: 10 Given:

```
public class Test {
public enum Dogs {collie, harrier, shepherd};
public static void main(String [] args) {
Dogs myDog = Dogs.shepherd;
switch (myDog) {
case collie:
```

```
System.out.print("collie ");
case default:
System.out.print("retriever ");
case harrier:
System.out.print("harrier ");
}
}
}
```

What is the result?

A. harrier
B. shepherd
C. retriever
D. Compilation fails.
E. retriever harrier
F. An exception is thrown at runtime.

Answer: D

Q: 11 Given:

```
Runnable r = new Runnable() {
public void run() {
System.out.print("Cat");
}
};
Thread t = new Thread(r) {
public void run() {
System.out.print("Dog");
}
};
t.start();
```

What is the result?

A. Cat
B. Dog
C. Compilation fails.
D. The code runs with no output.
E. An exception is thrown at runtime.

Answer: B

Q: 12 Given:

```
public class Threads4 {
public static void main (String[] args) {
new Threads4().go();
}
public void go() {
Runnable r = new Runnable() {
public void run() {
System.out.print("foo");
}
};
Thread t = new Thread(r);
t.start();
t.start();
}
}
```

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes normally and prints "foo".

D. The code executes normally, but nothing is printed.

Answer: B

Q: 13 Given:

```
1. public class TestFive {
2. private int x;
3. public void foo() {
4. int current = x;
5. x = current + 1;
6. }
7. public void go() {
8. for(int i = 0; i < 5; i++) {
9. new Thread() {
10. public void run() {
11. foo();
12. System.out.print(x + ", ");
13. } }.start();
14. } }
```

Which two changes, taken together, would guarantee the output: 1, 2, 3, 4, 5, ? (Choose two.)

A. move the line 12 print statement into the foo() method

B. change line 7 to public synchronized void go() {

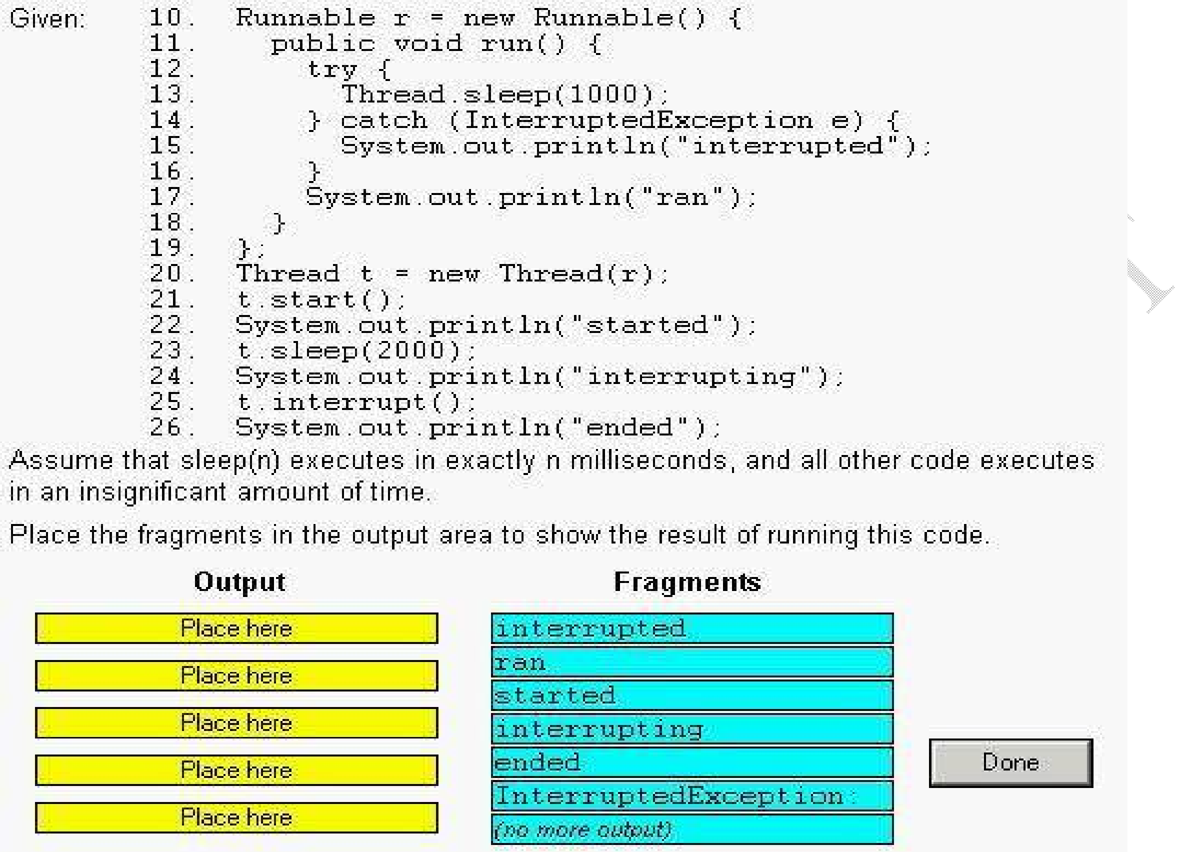
C. change the variable declaration on line 2 to private volatile int x;

D. wrap the code inside the foo() method with a synchronized(this) block

E. wrap the for loop code inside the go() method with a synchronized block synchronized(this)
{ // for loop code here }

Answer: A, D

Q: 14 Click the Task button.



Given:

```
Runnable r = new Runnable() {
  public void run() {
    try {
      Thread.sleep(1000);
    } catch (InterruptedException e) {
      System.out.println("interrupted");
    }
    System.out.println("ran");
  }
};
Thread t = new Thread(r);
t.start();
System.out.println("started");
t.sleep(2000);
System.out.println("interrupting");
t.interrupt();
System.out.println("ended");
```

Assume that sleep(n) executes in exactly n milliseconds, and all other code executes in an insignificant amount of time.

Place the fragments in the output area to show the result of running this code.

Output

Fragments

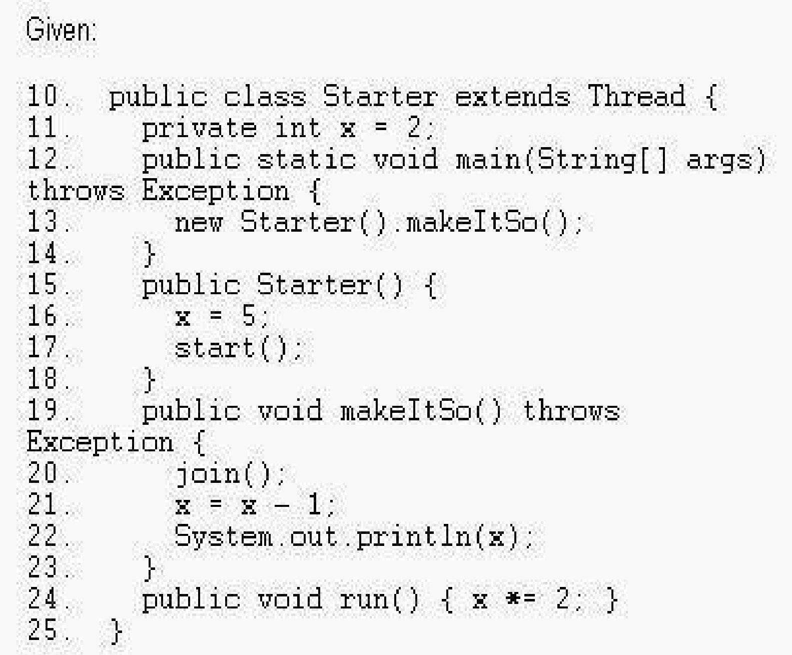
```
interrupted
ran
started
interrupting
ended
InterruptedException
```

(no more output)

Solution:
1.started
2.ran
3.interrupting
4.ended
5.(no more out put)

15 Click the Exhibit button.

What is the output if the main() method is run?



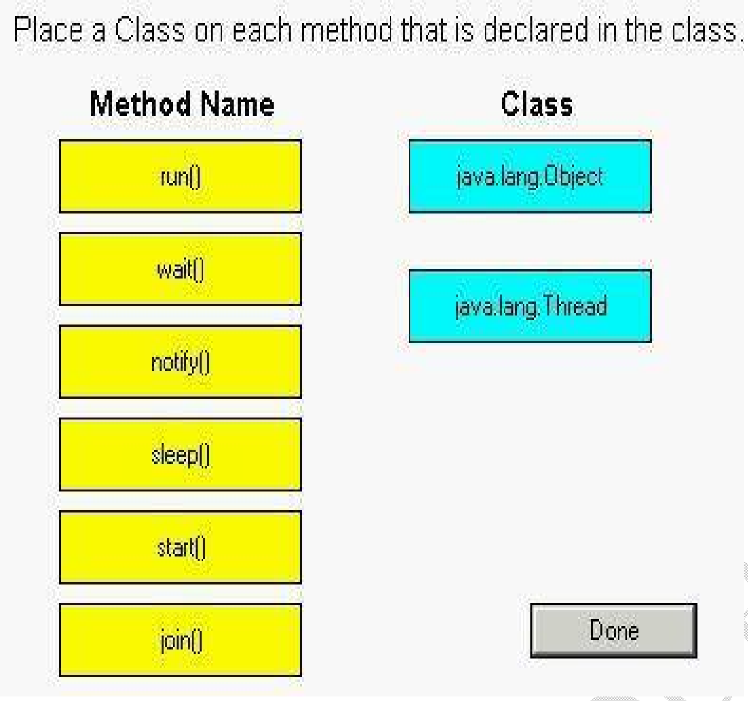
Given:

```
public class Starter extends Thread {
  private int x = 2;
  public static void main(String[] args) throws Exception {
    new Starter().makeItSo();
  }
  public Starter() {
    x = 5;
    start();
  }
  public void makeItSo() throws Exception {
    join();
    x = x - 1;
    System.out.println(x);
  }
  public void run() { x *= 2; }
}
```

A. 4 B. 5 C. 8
D. 9 E. Compilation fails.
F. An exception is thrown at runtime.
G. It is impossible to determine for certain.

Answer: D

Q: 16 Click the Task button.



Place a Class on each method that is declared in the class.

Solution:
java.lang.object ------------------------------ wait( ); notify( );
java.lang.Thread--------------------------------run(), sleep( ); start( ); join ( )

Q: 17 Given:
foo and bar are public references available to many other threads. foo refers to a Thread and bar is an
Object. The thread foo is currently executing bar.wait().
From another thread, what provides the most reliable way to ensure that foo will stop executing wait()?

A. foo.notify(); B. bar.notify();
C. foo.notifyAll(); D. Thread.notify();
E. bar.notifyAll(); F. Object.notify();

Answer: E

Q: 18 Given:

```
1. public class MyLogger {
2. private StringBuilder logger = new StringBuuilder();
3. public void log(String message, String user) {
4. logger.append(message);
5. logger.append(user);
6. }
7. }
```

The programmer must guarantee that a single MyLogger object works properly for a multi-threaded system.
How must this code be changed to be thread-safe?
A. synchronize the log method
B. replace StringBuilder with StringBuffer
C. replace StringBuilder with just a String object and use the string concatenation (+=) within the log method
D. No change is necessary, the current MyLogger code is already thread-safe.

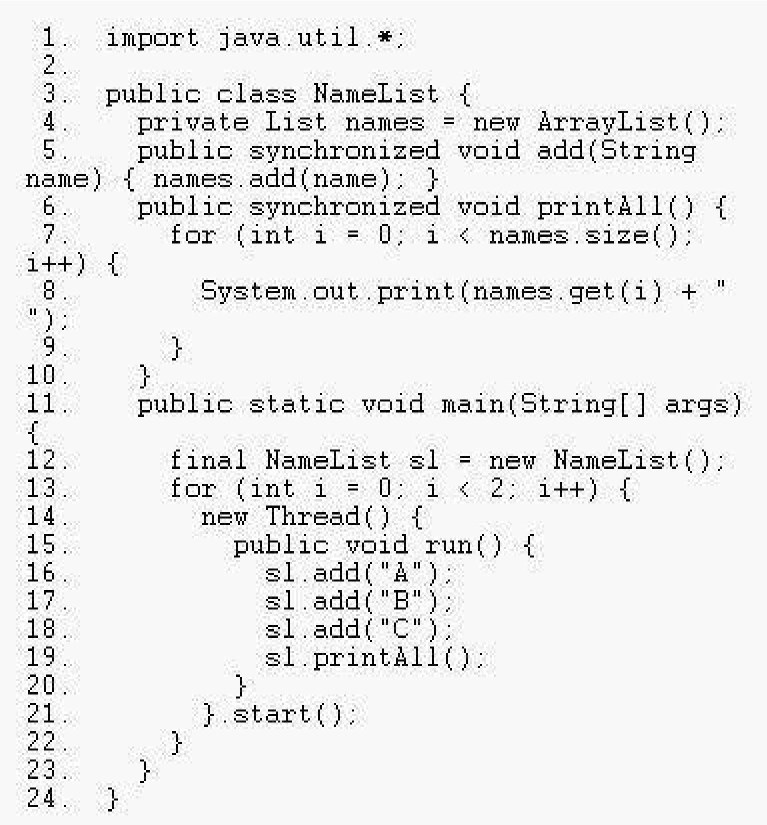
Answer: A

Q: 19 Given:

```
public class TestSeven extends Thread {
private static int x;
public synchronized void doThings() {
int current = x;
current++;
x = current;
}
public void run() {
doThings();
}
}
```

Which statement is true?
A. Compilation fails.
B. An exception is thrown at runtime.
C. Synchronizing the run() method would make the class thread-safe.
D. The data in variable "x" are protected from concurrent access problems.
E. Declaring the doThings() method as static would make the class thread-safe.
F. Wrapping the statements within doThings() in a synchronized(new Object()) { } block would make the class thread-safe.
Answer: E

Q:20 Click the Exhibit button.
Which two statements are true if this class is compiled and run? (Choose two.)



```
import java.util.*;

public class NameList {
  private List names = new ArrayList();
  public synchronized void add(String name) { names.add(name); }
  public synchronized void printAll() {
    for (int i = 0; i < names.size(); i++) {
      System.out.print(names.get(i) + " ");
    }
  }
  public static void main(String[] args) {
    final NameList sl = new NameList();
    for (int i = 0; i < 2; i++) {
      new Thread() {
        public void run() {
          sl.add("A");
          sl.add("B");
          sl.add("C");
          sl.printAll();
        }
      }.start();
    }
  }
}
```

A. An exception may be thrown at runtime.
B. The code may run with no output, without exiting.
C. The code may run with no output, exiting normally.
D. The code may run with output "A B A B C C ", then exit.
E. The code may run with output "A B C A B C A B C ", then exit.
F. The code may run with output "A A A B C A B C C ", then exit.
G. The code may run with output "A B C A A B C A B C ", then exit.

Answer: E, G

Q: 21 Given:

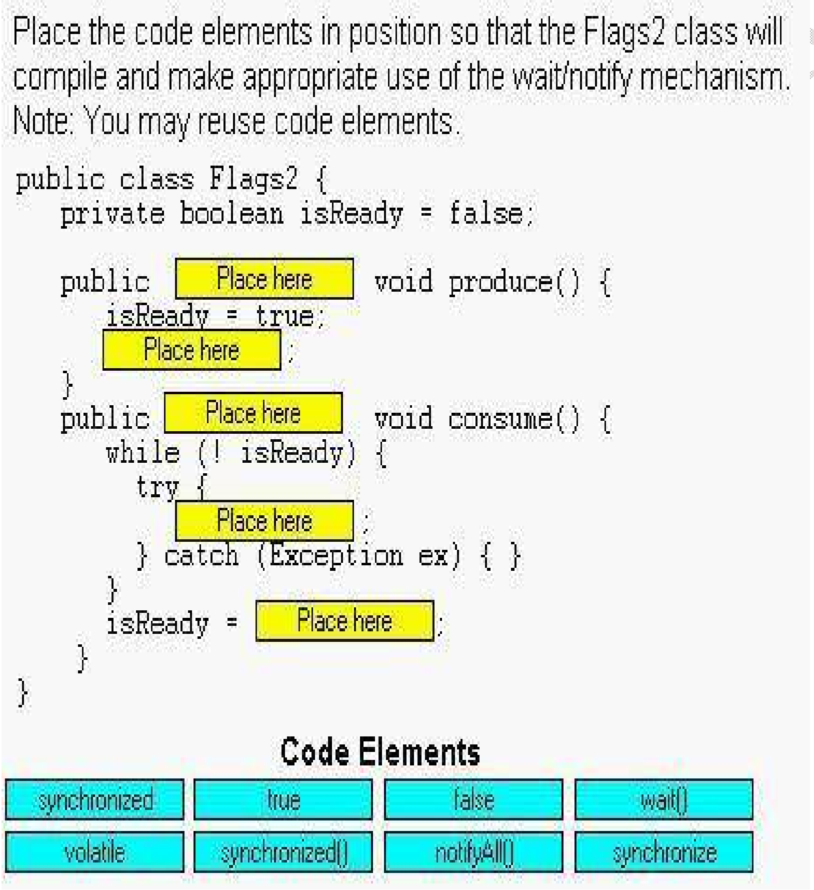
```
public class Threads5 {
public static void main (String[] args) {
new Thread(new Runnable() {
public void run() {
System.out.print("bar");
}}).start();
}
}
```

What is the result?
A. Compilation fails.

B. An exception is thrown at runtime.
C. The code executes normally and prints "bar".
D. The code executes normally, but nothing prints.
Answer: C

Q: 22 Which three will compile and run without exception? (Choose three.)
A. private synchronized Object o;
B. void go() {
synchronized() { /* code here */ }
C. public synchronized void go() { /* code here */ }
D. private synchronized(this) void go() { /* code here */ }
E. void go() {
synchronized(Object.class) { /* code here */ }
F. void go() {
Object o = new Object();
synchronized(o) { /* code here */ }
Answer: C, E, F

Q: 23 Click the Task button.



Place the code elements in position so that the Flags2 class will compile and make appropriate use of the wait/notify mechanism.
Note: You may reuse code elements.

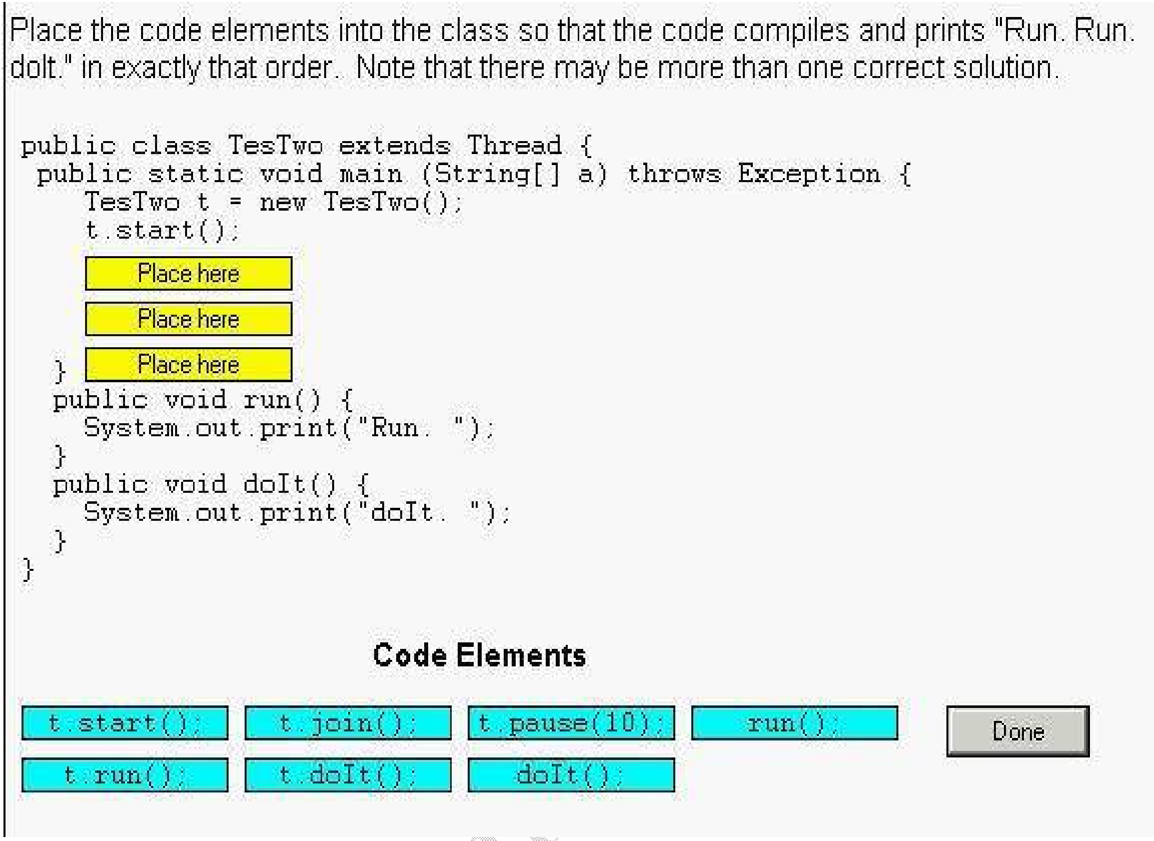
```
public class Flags2 {
   private boolean isReady = false;

   public  Place here  void produce() {
      isReady = true;
      Place here ;
   }
   public  Place here  void consume() {
      while (! isReady) {
        try {
          Place here ;
        } catch (Exception ex) { }
      }
      isReady =  Place here ;
    }
}
```

Code Elements

Solution:
1.synchronized
2.notifyAll( )
3.synchronized

4.wait( )
5.false

Q: 24 Click the Task button.



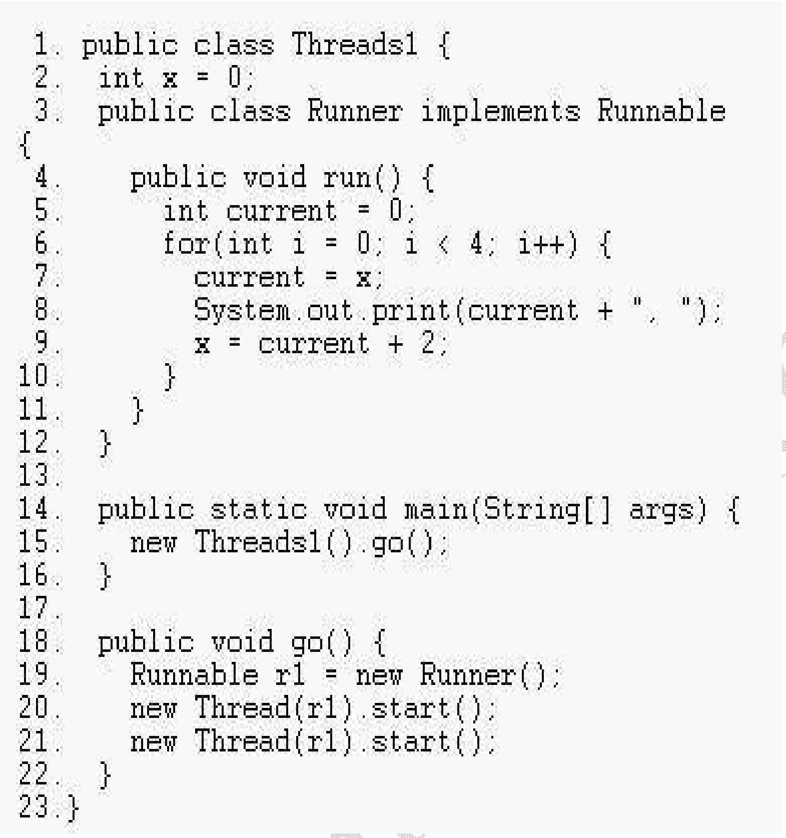
Place the code elements into the class so that the code compiles and prints "Run. Run. doIt." in exactly that order. Note that there may be more than one correct solution.

```
public class TesTwo extends Thread {
 public static void main (String[] a) throws Exception {
    TesTwo t = new TesTwo();
    t.start();
    Place here
    Place here
  } Place here
  public void run() {
    System.out.print("Run. ");
  }
  public void doIt() {
    System.out.print("doIt. ");
  }
}
```

Solution:

t.join( );
t.run( );
t.doIt( );

Q: 25 Click the Exhibit button.
Which two are possible results? (Choose two.)



```
public class Threads1 {
  int x = 0;
  public class Runner implements Runnable
{
    public void run() {
      int current = 0;
      for(int i = 0; i < 4; i++) {
        current = x;
        System.out.print(current + ", ");
        x = current + 2;
      }
    }
  }

  public static void main(String[] args) {
    new Threads1().go();
  }

  public void go() {
    Runnable r1 = new Runner();
    new Thread(r1).start();
    new Thread(r1).start();
  }
}
```

A. 0, 2, 4, 4, 6, 8, 10, 6,
B. 0, 2, 4, 6, 8, 10, 2, 4,
C. 0, 2, 4, 6, 8, 10, 12, 14,
D. 0, 0, 2, 2, 4, 4, 6, 6, 8, 8, 10, 10, 12, 12, 14, 14,
E. 0, 2, 4, 6, 8, 10, 12, 14, 0, 2, 4, 6, 8, 10, 12, 14,

Answer: A, C

Question: 26
Click the Exhibit button.
Given:
1**. public class TwoThreads {**
2
3. private static Object resource = new Object();
4.
5. private static void delay(long n) {
6. try { Thread.sleep(n); }
7. catch (Exception e) { System.out.print("Error "); }

```
}

public static void main(String[] args) {
System.out.print(”StartMain “);
new Thread1().start();
delay(1000);
Thread t2 = new Thread2();
t2.start();
delay(1000);
t2.interrupt
delay(1000);
System.out.print(”EndMain “);
}

static class Thread 1 extends Thread {
public void run() {
synchronized (resource) {
System.out.print(”Startl “);
delay(6000);
System.out.print(”End1 “);
}
}
}

static class Thread2 extends Thread {
public void run() {
synchronized (resource) {
System.out.print(”Start2 “);
delay(2000);
System.out.print(”End2 “);
}
}
}
}
```

Assume that sleep(n) executes in exactly m milliseconds, and all other code executes in an insignificant amount of time. What is the output if the main() method is run?

A. Compilation fails.
B. Deadlock occurs.
C. StartMain Start1 Error EndMain End1
D. StartMain Start1 EndMain End1 Start2 End2
E. StartMain Start1 Error Start2 EndMain End2 End1
F. StartMain Start1 Start2 Error End2 EndMain End1
G. StartMain Start1 EndMain End1 Start2 Error End2

Answer: G

Question: 27
Click the Exhibit button.

```
10. public class Transfers {
11. public static void main(String[] args) throws Exception {
12. Record r1 = new Record();
13. Record r2 = new Record();
14. doTransfer(r1, r2, 5);
15. doTransfer(r2, r1, 2);
16. doTransfer(r1, r2, 1);
17. // print the result
18. System.out.println(”rl = “ + r1.get() +“, r2=” + r2.get());
19. }
20. private static void doTransfer(
21. final Record a, final Record b, final int amount) {
22. Thread t = new Thread() {
23. public void run() {
24. new Clerk().transfer(a, b, amount);
25. }
26. };
27. t.start();
28. }
29. }
30. class Clerk {
31. public synchronized void transfer(Record a, Record b, int amount){
32. synchronized (a) {
33. synchronized (b) {
34. a.add(-amount);
35. b.add(amount);
36. }
37. }
38. }
39. }
40. class Record {
41.int num=10;
42. public int get() { return num; }
43. public void add(int n) { num = num + n; }
44. }
```

If Transfers.main() is run, which three are true? (Choose three.)

A. The output may be “r1 = 6, r2 = 14”.

B. The output may be “r1 = *5*, r2 = *15*”.

C. The output may be “r1 = 8, r2 = 12”.
D. The code may run (and complete) with no output.
E. The code may deadlock (without completing) with no output.
F. M IllegalStateException or InterruptedException may be thrown at runtime.
Answer: ABE

Question: 28
Which two statements are true? (Choose two.)
A. It is possible for more than two threads to deadlock at once.
B. The JVM implementation guarantees that multiple threads cannot enter into a deadlocked state.
C. Deadlocked threads release once their sleep() method's sleep duration has expired.
D. Deadlocking can occur only when the wait(), notify(), and notifyAll() methods are used incorrectly.
E. It is possible for a single-threaded application to deadlock if synchronized blocks are used incorrectly.
F. If a piece of code is capable of deadlocking, you cannot eliminate the possibility of deadlocking by inserting
invocations of Thread.yield().
Answer: A, F

Question: 29
Given:

```
class PingPong2 {
synchronized void hit(long n) {
for(int i = 1; i < 3; i++)
System.out.print(n + "-" + i + " ");
}
}
public class Tester implements Runnable {
static PingPong2 pp2 = new PingPong2();
public static void main(String[] args) {
new Thread(new Tester()).start();
new Thread(new Tester()).start();
}
public void run() { pp2.hit(Thread.currentThread().getId()); }
}
```

Which statement is true?
A. The output could be 5-1 6-1 6-2 5-2
B. The output could be 6-1 6-2 5-1 5-2
C. The output could be 6-1 5-2 6-2 5-1
D. The output could be 6-1 6-2 5-1 7-1
Answer: B

Question: 30
Given that t1 is a reference to a live thread, which is true?
A. The Thread.sleep() method can take t1 as an argument.
B. The Object.notify() method can take t1 as an argument.

C. The Thread.yield() method can take t1 as an argument.
D. The Thread.setPriority() method can take t1 as an argument.
E. The Object.notify() method arbitrarily chooses which thread to notify.
Answer: E

Question: 31
Given that Triangle implements Runnable, and:

```
31. void go() throws Exception {
32. Thread t = new Thread(new Triangle());
33. t.start();
34. for(int x = 1; x < 100000; x++) {
35. //insert code here
36. if(x%100 == 0) System.out.print("g");
37. } }
38. public void run() {
39. try {
40. for(int x = 1; x < 100000; x++) {
41. // insert the same code here
42. if(x%100 == 0) System.out.print("t");
43. }
44. } catch (Exception e) { }
45. }
```

Which two statements, inserted independently at both lines 35 and 41, tend to allow both threads to temporarily pause and allow the other thread to execute? (Choose two.)
A. Thread.wait();
B. Thread.join();
C. Thread.yield();
D. Thread.sleep(1);
E. Thread.notify();

Answer: C,D