Given:

```
13. public class Pass{
14. public static void main(String [] args){
15. int x = 5;
16. Pass p = new Pass();
17. p.doStuff(x);
18. System.out.println(“ main x =” + x);
19. }
20.
21. void doStuff(int x){
22. System.out.print(“ do Stuff x =” + x++);
23. }
24. }
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. doStuff x = 6 main x = 6
D. doStuff x = 5 main x = 5
E. doStuff x = 5 main x = 6
F. doStuff x = 6 main x = 5

En Java todos los valores se pasan por valor. En el caso de una primitiva, lo que le estoy pasando al método doStuff(int x) es el valor de la variable x, pero no la referencia a la primitiva x, luego el valor de la variable miembro x permanecerá inalterado. D

Given:

```
1. public class GC{
2. private Object o;
3. private void doSomethingElse(Object obj){o = obj}
4. public void doSomething(){
5. Object o = new Object();
6. doSomethingElse(o);
7. o = new Object();
8. doSomethingElse(null);
9. o = null;
10. }
11. }
```

When the doSomething method is called, after which line does the Object created in line 5 become avaiable for garbage collection?

A. Line 5
B. Line 6
C. Line 7
D. Line 8
E. Line 9
F. Line 10

En Java todos los valores se pasan por valor. En el caso de un objeto, lo que esto pasando es el valor de una referencia, luego cuando dicha referencia se iguale a null, afectará a la correspondiente variable miembro. D

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. }
public void run(){
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”.

El método join espera a que el hilo muera. E

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.

C

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

```
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.

D

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.println(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
}

A, D

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
```

D, E

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.

Método notify(). Wakes up a single thread that is waiting on this object's monitor. If any threads are waiting on this object, one of them is chosen to be awakened. The choice is arbitrary and occurs at the discretion of the implementation. A thread waits on an object's monitor by calling one of the `wait` methods. E

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 nad 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();

yield() → Causes the currently executing thread object to temporarily pause and allow other threads to execute.
Join() → Waits at most `millis` milliseconds for this thread to die. A timeout of `0` means to wait forever.
sleep(long millis) → Causes the currently executing thread to sleep (temporarily cease execution) for the specified number of milliseconds.
Notify() → Wakes up a single thread that is waiting on this object's monitor. If any threads are waiting on this object, one of them is chosen to be awakened. The choice is arbitrary and occurs at the discretion of the implementation.
C, D

Given:

```
1. public class Threads3 implements Runnable{
2. public void run(){
3. System.out.print("running");
4. }
5. public static void main (String [] args){
6. Thread t = new Thread(new Threads3());
7. t.run();
8. t.run();
9. t.start();
10. }
11. }
```

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”.

E

Given:
```
11. public class PingPong implements Runnable{
12. synchronized void hit(long n){
13. for(int i = 1; i<3;  i++)
14. System.out.print(n + “-” + i + “ ”);
15. }
16. public static void main(String [] args){
17. new Thread(new PingPong()).start();
18. new Thread(new PingPong()).start();
19. }
20. public void run(){
21. hit(Thread.currentThread().getId());
22. }
23. }
```
Which two statements are true?(Choose two.)
A. The output could be 8-1 7-2 8-2 7-1
B. The output could be 7-1 7-2 8-1 6-1
C. The output could be 8-1 7-1 7-2 8-2
D. The output could be 8-1 8-2 7-1 7-2

Para cada hilo, de acuerdo al código impuesto por el bucle for, los numeros irán aumentando de menos a más.C, D

Given:
```
1. public class TestOne{
2. public static void main(String [] args) throws Exception{
3. Thread.sleep(3000);
4. System.out.println(“sleep”);
5. }
6. }
```
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.

C

Which two stataments 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 posibility of deadlocking by inserting invocations of Thread.yield().

A, F

Given:
```
1. public class Threads4{
2. public static void main(String [] args){
3. new Threads4().go();
4. }
5. public void go(){
6. 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

Runnable es una interfaz y no se puede instanciar. B

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 male the class thread-safe.
D. The data 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.

E

Which two code fragments will execute the method doStuff() in a seperate 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();

D, F

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 safely by multiple threads?(Choose three.)
A. declare reset() using the synchronized keyword
B. declare getName() using synchronized keyword
C. declare getCount() using the synchronized keyword
D. delcare the constructor using the synchronized keyword
E. declare increment() using the synchronized keyword

A, C, E

What is the output if the main() method is run?
Given:
10. public class Starter extends Thread{
11. private int x = 2;
12. public static void main(String [] args) throws Excetpion{
13. new Starter().makeItSo();
14. }
15. public Starter(){
16. x = 5;
17. start();
18. }
19. public void makeItSo() throws Exception{
20. join();
21. x = x -1;
22. System.out.println(x);
23. }
24. public void run(){x*=2}
25.}
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.

D