Exam Number/Code : 1Z0-851

Exam Name: Java Standard Edition 6 Programmer Certified Professional Exam

Version : Demo

QUESTION 1
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: DE

Explanation/Reference:

```
End of method.
run.
Exception in thread "Thread-0" java.lang.RuntimeException: Problem
    at Threads2.run(Threads2.java:5)
    at java.lang.Thread.run(Unknown Source)
```

QUESTION 2
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: AF

QUESTION 3
Given:

```
void waitForSignal() {
    Object obj = new Object();
    synchronized (Thread.currentThread()) {
        obj.wait();
        obj.notify();
    }
}
```

Which statement is true?

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

Answer: B

Explanation/Reference:
Not quite sure about the answer, because first of all this code will not compile:

```
Threads2.java:15: unreported exception
java.lang.InterruptedException; must be caught or declared to be
thrown
            obj.wait();
                    ^
1 error
```

QUESTION 4
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 5
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

Explanation/Reference:

```
Exception in thread "main" java.lang.IllegalThreadStateException
    at java.lang.Thread.start(Unknown Source)
    at Threads4.go(Threads4.java:14)
    at Threads4.main(Threads4.java:3)
foo
```

QUESTION 6

Given:

```
public abstract class Shape {
    private int x;
    private int y;

    public abstract void draw();

    public void setAnchor(int x, int y) {
        this.x = x;
        this.y = y;
    }
}
```

Which two classes use the Shape class correctly? (Choose two.)

A.
```
public class Circle implements Shape {
    private int radius;
```

```
}
```

B.
```
public abstract class Circle extends Shape {
    private int radius;
}
```

C.
```
public class Circle extends Shape {
    private int radius;
    public void draw();
}
```

D.
```
public abstract class Circle implements Shape {
    private int radius;
    public void draw();
}
```

E.
```
public class Circle extends Shape {
    private int radius;
    public void draw() {/* code here */}
}
```

F.
```
public abstract class Circle implements Shape {
    private int radius;
    public void draw() {/* code here */}
}
```

Answer: BE

QUESTION 7

Given:

```
public class Barn {
    public static void main(String[] args) {
        new Barn().go("hi", 1);
        new Barn().go("hi", "world", 2);
    }

    public void go(String... y, int x) {
        System.out.print(y[y.length - 1] + " ");
    }
}
```

What is the result?

A. hi hi
B. hi world

C. world world
D. Compilation fails.
E. An exception is thrown at runtime.

Answer: D

Explanation/Reference:
The method go(String[], int) in the type Barn is not applicable for the arguments (String, int)

The variable argument type String of the method go must be the last parameter

QUESTION 8
Given:

```
class Nav{
    public enum Direction { NORTH, SOUTH, EAST, WEST }
}

public class Sprite{
    // insert code here
}
```

Which code, inserted at line 14, allows the Sprite class to compile?

A. Direction d = NORTH;
B. Nav.Direction d = NORTH;
C. Direction d = Direction.NORTH;
D. Nav.Direction d = Nav.Direction.NORTH;

Answer: D

QUESTION 9
Which statement is true about the classes and interfaces in the exhibit?

```
01. public interface A {
02.   public void doSomething(String thing);
03. }

01. public class AImpl implements A {
02.    public void doSomething(String msg) {}
```

```
03. }
```

```
01. public class B {
02.   public A doit(){
03.        //more code here
04.    }
05.    public String execute(){
06.       //more code here
07.   }
08. }
```

```
01. public class C extends B {
02.   public AImpl doit(){
03.        //more code here
04.    }
05.
06.    public Object execute() {
07.        //more code here
08.    }
09. }
```

A. Compilation will succeed for all classes and interfaces.
B. Compilation of class C will fail because of an error in line 2.
C. Compilation of class C will fail because of an error in line 6.
D. Compilation of class AImpl will fail because of an error in line 2.

Answer: C

Explanation/Reference:
The return type is incompatible with B.execute()

QUESTION 10
What is the result?

```
11. public class Person {
12.    String name = "No name";
13.    public Person(String nm) { name = nm; }
14. }
15.
16. public class Employee extends Person {
17.    String empID = "0000";
18.    public Employee(String id) { empID = id; }
19. }
```

```
20.
21. public class EmployeeTest {
22.    public static void main(String[] args){
23.        Employee e = new Employee("4321");
24.        System.out.println(e.empID);
25.    }
26. }
```

A. 4321
B. 0000
C. An exception is thrown at runtime.
D. Compilation fails because of an error in line 18.

Answer: D

Explanation/Reference:
Implicit super constructor Person() is undefined. Must explicitly invoke another constructor