DURGA ONLINE EXAMS

Test Your Knowledge

DURGAJOBS.COM
Continuous Job Updates For Every hour

HOME

191) **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?

1) Compilation fails.
2) An exception is thrown at runtime.
3) The code executes normally and prints "sleep".
4) The code executes normally, but nothing is printed.

Your Selected options	::	none ✗
Correct Options	::	3

Click Here for Explanation

192) **Which two code fragments will execute the method doStuff() in a separate thread? (Choose two.)**

1)
```
new Thread() {
public void run() { doStuff(); }
};
```
2)
```
new Thread() {
public void start() { doStuff(); }
};
```
3)
```
new Thread() {
public void start() { doStuff(); }
}.run();
```
4)
```
new Thread() {
public void run() { doStuff(); }
}.start();
```
5)
```
new Thread(new Runnable() {
public void run() { doStuff(); }
}).run();
```
6)
```
new Thread(new Runnable() {
public void run() { doStuff(); }
}).start();
```

Your Selected options	::	none ✗
Correct Options	::	4, 6

Click Here for Explanation

193) **Which three will compile and run without exception? (Choose three.)**

1) `private synchronized Object o;`
2)
```
void go() {
synchronized() { /* code here */ }
```
3) `public synchronized void go() { /* code here */ }`
4) `private synchronized(this) void go() { /* code here */ }`
5)
```
void go() {
synchronized(Object.class) { /* code here */ }
```
6)
```
void go() {
Object o = new Object();
synchronized(o) { /* code here */ }
```

Your Selected options :: none

Correct Options :: 3, 5, 6 ✗

Click Here for Explanation

194) **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.)

1) **move the line 12 print statement into the foo() method**

2) **change line 7 to public synchronized void go() {**

3) **change the variable declaration on line 2 to private volatile int x;**

4) **wrap the code inside the foo() method with a synchronized(this) block**

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

Your Selected options :: none ✗

Correct Options :: 1, 4

Click Here for Explanation

195) **Given:**
```
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(â€rl = â€œ + r1.get() +â€œ, r2=â€ + 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
```
1) ***The output may be â€œr1 = 6, r2 = 14â€.***

2) ***The output may be â€œr1 = 5, r2 = 15â€.***

3) ***The output may be â€œr1 = 8, r2 = 12â€.***

4) ***The code may run (and complete) with no output.***

5) ***The code may deadlock (without completing) with no output.***

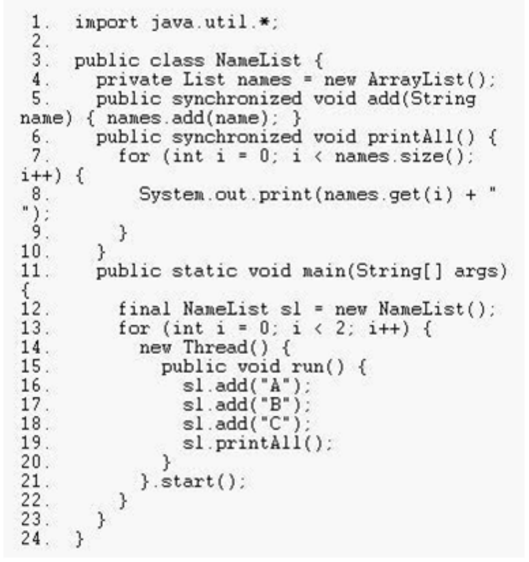
6) ***IllegalStateException or InterruptedException may be thrown at runtime.***

Your Selected options :: none ✗

Correct Options :: 1, 2, 5

Click Here for Explanation

196) **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();
    }
  }
}
```

1) **An exception may be thrown at runtime.**

2) **The code may run with no output, without exiting.**

3) **The code may run with no output, exiting normally.**

4) **The code may run with output "A B A B C C ", then exit.**

5) **The code may run with output "A B C A B C A B C ", then exit.**

6) **The code may run with output "A A A B C A B C C ", then exit.**

7) **The code may run with output "A B C A A B C A B C ", then exit.**

Your Selected options :: none

Correct Options :: 5, 7

Click Here for Explanation

197) **Given:**
1. public class TestSeven extends Thread {
2. private static int x;
3. public synchronized void doThings() {
4. int current = x;
5. current++;
6. x = current;
7. }
8. public void run() {
9. doThings();
10. }
11.}
Which statement is true?

1) **Compilation fails.**

2) **An exception is thrown at runtime.**

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

4) **The data in variable "x" are protected from concurrent access problems.**

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

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

Your Selected options :: none

Correct Options :: 5

Click Here for Explanation

198) **Given this method in a class:**
21. public String toString() {

22. StringBuffer buffer = new StringBuffer();
23. buffer.append('<');
24. buffer.append(this.name);
25. buffer.append('>');
26. return buffer.toString();
27. }
Which statement is true?

1) **This code is NOT thread-safe.**

2) **The programmer can replace StringBuffer with StringBuilder with no other changes.**

3) **This code will perform poorly. For better performance, the code should be rewritten: return "<" + this.name + ">";**

4) **This code will perform well and converting the code to use StringBuilder will not enhance the performance.**

Your Selected options :: none ✗

Correct Options :: 2

Click Here for Explanation

199) **Which two scenarios are NOT safe to replace a StringBuffer object with a StringBuilder object? (Choose two.)**

1) **When using versions of Java technology earlier than 5.0.**

2) **When sharing a StringBuffer among multiple threads.**

3) **When using the java.io class StringBufferInputStream.**

4) **When you plan to reuse the StringBuffer to build more than one string.**

Your Selected options :: none ✗

Correct Options :: 1, 2

Click Here for Explanation

200) **Given:**
10. public class MyClass {
11.
12. public Integer startingI;
13. public void methodA() {
14. Integer i = new Integer(25);
15. startingI = i;
16. methodB(i);
17. }
18. private void methodB(Integer i2) {
19. i2 = i2.intValue();
20.
21. }
22. }
If methodA is invoked, which two are true at line 20? (Choose two.)

1) **i2 == startingI returns true.**

2) **i2 == startingI returns false.**

3) **i2.equals(startingI) returns true.**

4) **i2.equals(startingI) returns false.**

Your Selected options :: none ✗

Correct Options :: 2, 3

Click Here for Explanation

« Prev | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | **20** | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | Next »

Total No.of Questions	::	292
Total No.of Answered Questions	::	0
Total No.of Unanswered Questions	::	292
Marks	::	0/292(0%)