Question 1 What is the result?

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(); //it will call the main thread not the new thread

t.run(); // and execute the main run() as it is overriden

t.start(); // now the new thread is created by start() method

// and it’s up to thread schedule to decide when to execute

}

}

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”.**

//Can we set the priority using thread constructor: no : 1-10 is valid parameters and only by priority method we can set the priority of thread

Question 2 : Which two code fragments will execute the method doStuff() in a separate thread? (Choose two.)

A. new Thread() {

public void run() { doStuff(); }

} // thread is created but not called

B. new Thread() {

public void start() { doStuff(); }

}

// thread is not created because start method is overridden. It will never advice to override the start method

C. new Thread() {

public void start() { doStuff(); }

} .run();

// thread is not created because start method is overridden. run() will have no effect.

**D. new Thread() {**

**public void run() { doStuff(); }**

**} .start();**

E. new Thread(new Runnable() {

public void run() { doStuff(); }

}).run();

// it will not create another thread instead it will call the run() method of main thread

**F. new Thread(new Runnable() {**

**public void run() { doStuff(); }**

**}).start();**

Question 3: What is the result?

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();

}

}

A. Compilation fails.

**B. An exception is thrown at runtime. java.lang.IllegalThreadStateException**

C. The code executes normally and prints ‘foo”. D. The code executes normally, but nothing is printed.

Question 4: What is the result?

public class Threads5 {

public static void main (String[] args) {

new Thread(new Runnable() {

public void run() {

System.out.print(”bar”);

}}).start();

}

}

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.

Question 5: What is the result?

Runnable r = new Runnable() {

public void run() {

System.out.print(”Cat”);

}

};

Thread t = new Thread(r) {

public void run() { //overriding the run method of the

System.out.print(”Dog”); //runnable interface

}

};

t.start();

A. Cat

**B. Dog**

C. Compilation fails.

D. The code runs with no output.

E. An exception is thrown at runtime.

Question 6 Click the Exhibit button. What is the output if the main() method is rum?

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.

Question 7: Which two can be results? (Choose two.)

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.”);

}

}

1. 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.**

Question 8

public class TestOne {

public static void main (String[] args) throws Exception {

Thread.sleep(3000);

System.out.println(”sleep”);

}

}

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.

Question 9:

public class TestOne implements Runnable {

public static void main (String[] args) throws Exception {

Thread t = new Thread(new TestOne());

t.start();

System.out.print(”Started”);

t.join();

System.out.print(”Complete”);

}

public void run() {

for (int i= 0; i< 4; i++) {

System.out.print(i);

}

}

}

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 “Started0l23Complete”.**

**Question 10:**

public class TwoThreads {

private static Object resource = new Object();

private static void delay(long n) {

try { Thread.sleep(n);

}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);

//interrupt method is called when you are in sleping mode

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(m) 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

Question 11: Which three changes should be made to adapt this class to be used safely by multiple threads? **(Choose three.)**

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() { coount++; }

public int getCount() { return count; }

public void reset() { count = 0; }

}

**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**

Question 12

1. public class Threads 1 {

2. intx=0;

3. public class Runner implements Runnable {

4. public void run() {

5. int current = 0;

6. for(int=i=0;i<4;i++){

7. current = x;

8. System.out.print(current + “, “);

9. x = current + 2;

10. }

11. }

12. }

13.

14. public static void main(String[] args) {

15. new Threads1().go();

16. }

17.

18. public void go() {

19. Runnable r1 = new Runner();

20. new Thread(r1).start();

21. new Thread(r1 ).start();

22. }

23. }

Which two are possible results? (Choose two.)

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,

Question 13: Which two statements are true if this class is compiled and run? (Choose two.)

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 rum with output “A B A B C C “, then exit.

E. The code may rum with output “A B C A B C A B C “, then exit.

F. The code may ruin with output “A A A B C A B C C “, then exit.

G. The code may ruin with output “A B C A A B C A B C “, then exit.

Question 14: Which two changes, taken together, would guarantee the output: 1, 2, 3, 4, 5,? Choose two

public class TestFive {

private int x;

public void foo() {

int current = x;

x = current + 1;

}

public void go() {

for(int i=0;i<5;i++) {

new Thread() {

public void run() {

foo();

System.out.print(x + “, “);

}

}.start();

}

}

}

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 3 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 }.

Question 15: 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 \*/ }

}