1 Which is the correct way to start a new thread?

Select the one correct answer.

(a) Just create a new `Thread` object. The thread will start automatically.

(b) Create a new `Thread` object and call the method `begin()`.

(c) Create a new `Thread` object and call the method `start()`.

(d) Create a new `Thread` object and call the method `run()`.

(e) Create a new `Thread` object and call the method `resume()`.

2 When extending the Thread class to implement the code executed by a thread, which method should be overridden?

Select the one correct answer.

(a) begin()

(b) start()

(c) run()

(d) resume()

(e) behavior()

3 Which statements are true?

Select the two correct answers.

(a) The class Thread is abstract.

(b) The class Thread implements Runnable.

(c) The Runnable interface has a single method named start.

(d) Calling the method run() on an object implementing Runnable will create a new thread.

(e) A program terminates when the last user thread finishes.

4 What will be the result of attempting to compile and run the following program?

```
public class MyClass extends Thread {
public MyClass(String s) { msg = s; }
String msg;
public void run() {
System.out.println(msg);
}
public static void main(String[] args) {
new MyClass("Hello");
new MyClass("World");
}
}
```

Select the one correct answer.

(a) The program will fail to compile.

(b) The program will compile without errors and will print Hello and World, in that order, every time the program is run.

(c) The program will compile without errors and will print a never-ending stream of Hello and World.

(d) The program will compile without errors and will print Hello and World when run, but the order is unpredictable.

(e) The program will compile without errors and will simply terminate without any output when run.

5 What will be the result of attempting to compile and run the following program?

```
class Extender extends Thread {
public Extender() { }
public Extender(Runnable runnable)
{super(runnable);}
public void run() {System.out.print("|Extender|");}
}

public class Implementer implements Runnable {
public void run()
{System.out.print("|Implementer|");}

public static void main(String[] args) {
new Extender(new Implementer()).start(); // (1)
new Extender().start(); // (2)
new Thread(new Implementer()).start(); // (3)
}
}
```

Select the one correct answer.

(a) The program will fail to compile.

(b) The program will compile without errors and will print |Extender| twice and |Implementer| once, in some order, every time the program is run.

(c) The program will compile without errors and will print|Extender| once and |Implementer| twice, in some order, every time the program is run.

(d) The program will compile without errors and will print |Extender| once and |Implementer| once, in some order, every time the program is run

(e) The program will compile without errors and will simply terminate without any output when run.

(f) The program will compile without errors, and will print |Extender| once and |Implementer| once, in some order, and terminate because of an runtime error.

6 What will be the result of attempting to compile and run the following program?

```
class R1 implements Runnable {
public void run() {
System.out.print(Thread.currentThread().getNa
me());
}
}
public class R2 implements Runnable {
public void run() {
new Thread(new R1(),"|R1a|").run();
new Thread(new R1(),"|R1b|").start();
System.out.print(Thread.currentThread().getNa
me());
}
public static void main(String[] args) {
new Thread(new R2(),"|R2|").start();
}
}
```

Select the one correct answer.

(a) The program will fail to compile.

(b) The program will compile without errors and will print |R1a| twice and |R2| once, in some order, every time the program is run.

(c) The program will compile without errors and will print|R1b| twice and |R2| once, in some order, every time the program is run.

(d) The program will compile without errors and will print |R1b| once and |R2| twice, in some order, every time the program is run.

(e) The program will compile without errors and will print |R1a| once, |R1b| once, and |R2| once, in some order, every time the program is run.

7 What will be the result of attempting to compile and run the following program?

```
public class Threader extends Thread {
Threader(String name) {
super(name);
}
public void run() throws IllegalStateException {
System.out.println(Thread.currentThread().getN
ame());
throw new IllegalStateException();
}
public static void main(String[] args) {
new Threader("|T1|").start();
}
}
```

Select the one correct answer.

(a) The program will fail to compile.

(b) The program will compile without errors, will print |T1|, and terminate normally every time the program is run.

(c) The program will compile without errors, will print|T1|, and throw an IllegalStateException, every time the program is run.

(d) None of the above.

8 What will be the result of attempting to compile and run the following program?

```
public class Worker extends Thread {
public void run() {
System.out.print("|work|");
}
public static void main(String[] args) {
Worker worker = new Worker();
worker.start();
worker.run();
worker.start();
}
}
```

Select the one correct answer.

(a) The program will fail to compile.

(b) The program will compile without errors, will print |work| twice, and terminate normally every time the program is run.

(c) The program will compile without errors, will print|work| three times, and terminate normally every time the program is run.

(d) The program will compile without errors, will print|work| twice, and throw an IllegalStateException, every time the program is run.

(e) None of the above.

9 Given the following program, which statements are guaranteed to be true?

```
public class ThreadedPrint {
static Thread makeThread(final String id,
boolean daemon) {
Thread t = new Thread(id) {
public void run() {
System.out.println(id);
}
};
t.setDaemon(daemon);
t.start();
return t;
}
public static void main(String[] args) {
Thread a = makeThread("A", false);
Thread b = makeThread("B", true);
System.out.print("End\n");
}
}
```

Select the two correct answers.

(a) The letter A is always printed.

(b) The letter B is always printed.

(c) The letter A is never printed after End.

(d) The letter B is never printed after End.

(e) The program might print B, End, and A, in that order.

10 Given the following program, which alternatives would make good choices to synchronize on at (1)?

```
public class Preference {
private int account1;
private Integer account2;
public void doIt() {
final Double account3 = new Double(10e10);
synchronized(/* ___(1)___ */) {
System.out.print("doIt");
}
}
}
```

Select the two correct answers.

(a) Synchronize on account1.

(b) Synchronize on account2.

(c) Synchronize on account3.

(d) Synchronize on this.

11 Which statements are not true about the synchronized block?

Select the three correct answers.

(a) If the expression in a synchronized block evaluates to null, a NullPointer-Exception will be thrown.

(b) The lock is only released if the execution of the block terminates normally.

(c) A thread cannot hold more than one lock at a time.

(d) Synchronized statements cannot be nested.

(e) The braces cannot be omitted even if there is only a single statement to execute in the block.

12 Which statement is true?

Select the one correct answer.

(a) No two threads can concurrently execute synchronized methods on the same object.

(b) Methods declared synchronized should not be recursive, since the object lock will not allow new invocations of the method.

(c) Synchronized methods can only call other synchronized methods directly.

(d) Inside a synchronized method, one can assume that no other threads are currently executing any other methods in the same class.