

1. **(1 point)** There are two types of exceptions in Java, checked exceptions, and unchecked exceptions. If a statement might throw a checked exception, the exception must either be caught, by placing the statement in a try-catch statement, or the method containing the statement must include a throws clause for the exception. How can you (or the compiler) tell if an exception is a checked, or unchecked exception?

Answer: Unchecked exceptions are the exception classes Error, RuntimeException, and any of their subclasses. Everything else is a checked exception.

2. **(2 points)** Given that SomeException and SomeOtherException are both checked exceptions, what must be true about the class SomeOtherException in the example below, in order for the classes to compile without errors?

Answer: SomeOtherException must be a subclass SomeException.

```
class Parent {
    void someMethod(int x) throws SomeException {
        if (x == 0) throw new SomeException("Some exception");
    }
}
class Child extends Parent {
    void someMethod(int x) throws SomeOtherException {
        if (x == 0) throw new SomeOtherException("Some other");
    }
}
```

3. **(1 point)** Assuming SomeOtherException is defined so that the classes in the previous question compile without errors, fill in the blank below so that this class also compiles without errors. Be as precise as possible. For example entering Exception in the blank would make the program compile, but this is not the best or most precise name to enter in the blank.

```
class Quiz5 {
    public static void main(String[] args) throws ___SomeException_____ {
        Parent p = new Child();
        p.someMethod(2);
    }
}
```

4. **(2 points)** The standard Java I/O classes use two techniques for detecting end-of-file, describe them. (The method tio.ReadInput.hasMoreElements() is a third technique, but not one supported by the standard Java I/O classes.)

Answer: If possible, a special value can be returned from input methods to indicate end-of-file. For example readLine() returns null. If no special value is possible (e.g. readInt()), then the input method throws an EOFException.

5. **(2 points)** When you use writeInt() from DataOutputStream, how many bytes are written for the value 123456789? Answer: 4

How many bytes are written for the same value if you use print() from PrintWriter? Answer: 9

6. **(1 point)** Synchronization in Java is done at the **object** level.
A. class B. object C. method D. variable

7. **(1 point)** What method in a Thread is similar to main() for a program? Answer run().

8. **(1 point)** To get a thread to execute concurrently with the rest of the program you call what method?
Answer: start()

9. **(2 points)** What are the possible outputs from the following program? Circle the letter for all outputs that are possible.

```
class Quiz5Prob9 {
    public static void main(String[] args) {
        Object obj = new Object();
        SomeThread t1 = new SomeThread(0, obj);
        SomeThread t2 = new SomeThread(1, obj);
        t1.XXX(); // Assume XXX was replaced with the answer to question 8.
        t2.XXX();
    }
}
class SomeThread extends Thread {
    private int id;
    Object lock;
    SomeThread(int id, Object lock) {
        this.id = id;
        this.lock = lock;
    }
    public void YYY() { // Assume YYY was replaced with the answer to question 7.
        synchronized(lock) {
            System.out.println(id + " starting."); // note this is inside the synchronized statement
            try {
                if (id == 0)
                    lock.wait();
                else
                    lock.notify();
            }
            catch (InterruptedException e) { }
        }
        System.out.println(id + " exiting."); // note this is outside the synchronized statement
    }
}
```

- | | | |
|---|---|---|
| A. 0 starting
1 starting
0 exiting
1 exiting | C. 0 starting
0 exiting
1 starting
1 exiting | E. 1 starting
1 exiting
0 starting
0 exiting |
| B. 1 starting
0 starting
1 exiting
0 exiting | D. 0 starting
1 starting
1 exiting
0 exiting | |

Answer: A and D