

Study questions set #4

1. State the full term abbreviated by “JIT” and define it in your own words.
2. Explain the difference between “checked” exceptions and “unchecked” exceptions in your own words.
3. Suppose the two class definitions below are in the same package. Implement the thrower() method as described in the comment.

```
public class MyException extends Exception {

    int code;

    public MyException(int code) {
        this.code = code;
    }

    public String getMessage() {
        return "Error code "+code;
    }

}

public class MyObj {

    int data;

    public MyObj(int data) {
        this.data = data;
    }

    /**
     * If data is even, throws a new MyException, whose error code
     * is equal to data.
     * @throws MyException
     */
    public void thrower() throws MyException {

    }

}
```

4. State the full console output from the following main() method. You may assume it is defined in the same package as MyObj and MyException.

```
public static void main(String[] args) {  
  
    MyObj obj1 = new MyObj(0);  
    MyObj obj2 = obj1;  
    MyObj obj3 = obj2;  
  
    try {  
  
        obj3.thrower();  
        System.out.println(obj3.data++);  
  
    } catch(MyException e1) {  
  
        try {  
  
            obj2.data = e1.code;  
            obj1.thrower();  
  
        } catch(MyException e2) {  
  
            System.out.println(e2.getMessage());  
  
        }  
  
    }  
  
}
```

5. Suppose the following unit tests are also in the same package as MyObj. For each of the test methods, indicate whether it passes, fails (due to a false assertion), or contains an error (compile-time or run-time). You may assume any necessary imports.

```
public class PracticeTests {

    // PASS _____ FAIL _____ ERROR _____
    @Test
    public void testOne() {
        MyObj obj1 = null;
        try {
            obj1.data = 3;
            assertTrue(true);
        } catch (Exception e) {
            assertTrue(false);
        }
    }

    // PASS _____ FAIL _____ ERROR _____
    @Test
    public void testTwo() {
        try {
            throw new RuntimeException();
        } catch (NullPointerException e) {
            assertTrue(true);
        } catch (Exception e) {
            assertTrue(false);
        }
    }

    // PASS _____ FAIL _____ ERROR _____
    @Test
    public void testThree() {
        boolean maybe = true;
        if(maybe = false) assertTrue(maybe);
        else assertTrue(maybe);
    }

}
```