1. (2 pts) When an error occurs during the execution of a program, we say that an exception has been thrown, and the hierarchy of classes that deal with these problems begins with the class Throwable in the java.lang package.

2. (2 pts) State Java’s catch or declare rule.

If a method has the potential for causing a checked exception to occur, then we must either list that checked exception in the throws clause of the method or create an exception handler to catch and deal with the exception.

3. (2 pts) Error, RuntimeException, and their subclasses are known as unchecked exceptions, and all other exceptions are known as checked exceptions.

4. (2 pts) Consider the following program. Apart from any error messages, what is printed when the program is executed? The output of the program is explained because Java supports the termination model of exception handling.

```java
import java.awt.*;
import javax.swing.*;
public class Question4 {
    private String string;
    public Question4() {
        string = "1";
    }
    public static void print() {
        System.out.println(1);
        System.out.println(string.length());
        System.out.println(2);
    }
    public static void main(String[] args) {
        Question4 question4 = new Question4();
        question4.print();
    }
}
```

Note: The following explanation is correct if you make print an instance variable. Only 1 will be printed. When an instance of Question4 is created, the constructor defines a local variable with the same name as the instance variable, string. So the instance variable has the initial value null. So in the print method, trying to call the method length() on string will cause a NullPointerException and so System.out.println(2) will not be executed. If you make string a static variable, then there will be still be an exception when the print method is executed so only 1 will print.

5. (2 pts) If ArithmeticException is a subclass of RuntimeException, and IOException is not a subclass of RuntimeException nor of Error, what are two compile-time errors in the following code?

```java
try {
```
One compile-time error is that the catch block for ArithmeticException is listed after the catch block for RuntimeException. The catch block for a subclass can’t be listed after the catch block for a superclass.

The other compile-time error is that since there is no line in the try block which could cause an IOException, which is a checked exception since it is not a subclass of RuntimeException nor of Error, and so we can’t have a catch block associated with that try block that catches an IOException.