CS 201, Fall 2013 — Oct 2nd — Exam 1 Name: _____

Question 1. [5 points] What output is printed by the following program (which begins on the left and continues on the right)?

```
public class Q1 {
  public int x;

  public Q1(int x) {
    this.x = x;
  }
```

```
  public static void main(String[] args) {
    Q1 a = new Q1(17);
    Q1 b = new Q1(39);

    b.x = a.x;
    a.x = 44;
    System.out.printf("%d,%d\n", a.x, b.x);
  }
}
```

Question 2. [5 points] What output is printed by the following program?

```
public class Q2 {
  public static void main(String[] args) {
    int[] a = new int[1];
    int[] b;

    a[0] = 99;
    b = a;
    b[0] = 101;
    System.out.printf("%d,%d\n", a[0], b[0]);
  }
}
```

Question 3. [10 points] Complete the following method. It should return the average of the minimum and maximum of the values stored in the `values` array. You may assume that the array will have at least one element.

```
public static double avgMinAndMax(double[] values) {
```

Question 4. [5 points] What output is printed by the following program (which begins on the left and continues on the right)?

```
public class Q4 {
  public static void f(int x)
    throws Exception {

    if (x % 2 == 0) {
      throw new Exception();
    }
    System.out.printf(
      "f(%d)\n", x);
  }
```

```
  public static void main(String[] a) {
    int[] values = { 1, 2, 3, 4 };
    for (int v : values) {
      try {
        f(v);
      } catch (Exception e) {
        System.out.printf("Exception!\n");
      }
    }
  }
}
```

Question 5. [10 points] Consider the following program:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class Q5 {
  public static void main(String[] a) throws IOException {
    Scanner keyboard = new Scanner(System.in);
    System.out.print("What file? ");
    String fileName = keyboard.nextLine();

    BufferedReader r = new BufferedReader(new FileReader(fileName));
    while (r.readLine() != null) {
      System.out.println(r.readLine());
    }

    r.close();
  }
}
```

(a) In words, explain what happens when this program is executed. Assume that the user types the name of a text file that does exist and is readable.

(b) Is the program guaranteed to close the `BufferedReader` if the file is opened successfully? Why or why not? Explain briefly.

Question 6. [10 points] Consider the following JUnit test class (which begins on the left and continues on the right):

```
public class ComboLockTest {
  private ComboLock lock;

  @Before
  public void setUp() {
    lock = new ComboLock(21, 8, 14);
  }
```

```
  @Test
  public void testInvalidCombo() {
    lock.spin(21);
    lock.spin(9);
    lock.spin(14);
    assertFalse(lock.isUnlocked());
  }

  @Test
  public void testValidCombo() {
    lock.spin(21);
    lock.spin(8);
    lock.spin(14);
    assertTrue(lock.isUnlocked());
  }
}
```

Show how the `ComboLock` class would be defined. You should show all of the public methods, but you do **not** need to show how the methods would be implemented. (Just leave the body of each method empty.)

Question 7. [15 points] Consider the following `Animal` class and partially-specified `FruitBat` class:

```
public abstract class Animal {
  private String sound;

  public Animal(String s) {
    this.sound = s;
  }

  public String getSound() {
    return sound;
  }

  public abstract
  boolean eatsInsects();
}
```

```
public class FruitBat [Missing code 1] {
  public FruitBat() {
    [Missing code 2]
  }

  [Missing code 3]
}
```

Now, consider the following statements:

```
Animal fruitBat = new FruitBat();
System.out.println(fruitBat.getSound());
System.out.println(fruitBat.eatsInsects());
```

Specify code that can be substituted for each of the three *Missing code* blocks that will allow these statements to compile and run successfully and print the output

```
Squeak
true
```

Use the other side of the page if necessary.

Programming Question

To get started, use a web browser to download the zipfile as specified by your instructor. Import it as an Eclipse project using File → Import... → General → Existing Projects into Workspace → Archive file.

Important: You may use the following resources:

- The lecture notes posted on the course web page
- Your previous labs and assignments

Do not open any other files, web pages, etc.

Question 8. [40 points] Complete the implementation of the `ComboLock` class. This class represents a combination lock, and should have each method described below.

Constructor: Takes three integer arguments, representing the numbers in the `ComboLock`'s combination.

`spin`: "Spins" the `ComboLock`. Takes a single integer parameter. If `spin` is called three times, and the sequence of values passed match the combination set when the `ComboLock` object was created, then the lock is *unlocked*.

`isUnlocked`: Checks whether the `ComboLock` is unlocked (as explained in the description for the `spin` method above). Returns `true` if the `ComboLock` is unlocked, or `false` if it is locked. This method does not take any parameters.

`lock`: "Resets" the combination lock by negating the effects of any previous calls to `spin`. In other words, after `lock` is called, the only way to unlock the `ComboLock` is to call `spin` three times, passing the sequence of numbers matching the combination. This method does not take any parameters.

Hints and requirements:

- In addition to adding the methods described above, you will need to add fields to keep track of the state of the `ComboLock`.
- The `ComboLockTest` JUnit test class can be used to test your implementation. Note that this class doesn't test the `lock` method: you should write at least one test for this method to make sure your `lock` method works correctly.
- Make sure that the `isUnlocked` method returns true only when the `spin` method has been called three times (following the creation of the `ComboLock` or a call to `lock`), passing the sequence of numbers matching the combination.

When you are ready to submit your code, select the **CS201_Exam1** project in the package explorer, and click the blue up arrow button in the toolbar. Enter your Marmoset username and password when prompted.