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

```
public class Q1 {                          public static void main(
  private int value;                           String[] args) {
                                             Q1 a = new Q1(17);
  public Q1(int v) {                         Q1 b = new Q1(42);
    value = v;                               System.out.printf("%d,%d\n",
  }                                            a.value, b.value);

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

Question 2. [5 points] What output is printed by the following code (which begins on the left and continues on the right)?

```
public class Q2 {                          public static void main(
  private int value;                           String[] args) {
                                             Q2 a = new Q2(17);
  public Q2(int v) {                         Q2 b = new Q2(42);
    value = v;
  }                                          mystery(a, b);

  public static void mystery(                System.out.printf("%d,%d\n",
      Q2 x, Q2 y) {                            a.value, b.value);
    Q2 tmp = x;                            }
    x = y;
    y = tmp;                             }
  }
```

Question 3. [10 points] Write statements to read two sets of (x,y) coordinates from the user, calculate the distance between the two sets of coordinates, and print the result. The distance between two points is $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. You can use `Math.sqrt` to compute the square root of a `double` value.

Question 4. [10 points] Identify, explain, and correct all of the problems with the following class definition, which begins on the left and continues on the right.

```
public class Point {                    private getX() {
   double x;                               return(x);
   double y;                            }

   Point() {                           private getY() {
     x = 0;                               return(y);
     y = 0;                            }
   }                                }

   Point(x,y) {
     x = x;
     y = y;
   }
```

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

```
public static int countLines(String fileName) {
  FileReader fr = new FileReader(fileName);
  BufferedReader br = new BufferedReader(fr);

  int count = 0;
  while (true) {
    String line = br.readLine();
    if (line == null) { break; }
    count++;
  }

  br.close();

  return count;
}
```

(a) At which lines in this method could an IOException or FileNotFoundException occur?

(b) As written, this method neither declares itself as throwing IOException nor attempts to handle IOException using try/catch. Which of these do you think would make more sense? Explain briefly.

(c) Explain how the method might open a file, but then not make an attempt to close it.

(d) Explain how to guarantee that the method will make an attempt to close the file if it is opened.

Question 6. [10 points] Implement the `CountBy` class so that the following JUnit tests will pass. The idea is that when a `CountBy` object is created, its internal counter is set to zero, and a parameter to the constructor sets an integer value that is the multiple by which the object will count. Each time the `next` method is called, the internal counter should be increased by the multiple. Each time the `get` method is called, it should return the object's current counter value.

```
CountBy by2 = new CountBy(2);
CountBy by7 = new CountBy(7);

assertEquals(0, by2.get());
assertEquals(0, by7.get());

by2.next();
assertEquals(2, by2.get());

by7.next();
by7.next();
assertEquals(14, by7.get());
```

Question 7. [10 points] For the `Point3D` class shown below, circle the fields and methods that a concrete class `Vector` extending `Point3D` would inherit from `Point3D`.

Also, list all the methods that subclass `Vector` has to instantiate to be considered a valid concrete class.

```
public abstract class Point3D {
  public    int x;
  public    int y;
  public    int z;

  public A() {
    x = 0;
    y = 0;
    x = 0;
  }

  public A(int x, int y, int z) {
    this.x = x;
    this.y = y;
    this.z = z;
  }

  public int getX() {
    return x;
  }

  public int getY() {
    return y;
  }

  public int getZ() {
    return z:
  }
```

```
  private int getX_2() {
    return x*x;
  }

  private int getY_2() {
    return y*y;
  }

  private int getZ_2() {
    return z*z;
  }

  public abstract double getLength();

  public abstract double getNormX();

  public abstract double getNormY();

  public abstract double getNormZ();
}

public class Vector
    extends Point3D {
  // what methods must
  // this class define?
}
```

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 textbook
- 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 `Histogram` class as follows:

- The constructor takes an `int` parameter, which specifies how many buckets the histogram should have
- Each bucket is an integer count
- The buckets are numbered 0 to $n - 1$, where n is the number of buckets
- The initial value of each bucket is 0
- The `increment` method takes a bucket number as a parameter, and increases the specified bucket's count by 1
- The `get` method takes a bucket number as a parameter, and returns the specified bucket's count

Hint: Use an array to keep track of the count for each bucket. The constructor should create the array, and it should store a reference to the array in a field so that it can be used in the `increment` and `get` methods.

A `HistogramTest` JUnit test class is provided. Make sure that all of the tests pass.

You can also run `HistogramGUI` as a visual test of your `Histogram` class. (But make sure that you do run the tests!)

When you are ready to submit your code, export the **CS201_Exam01** project as a zip file and upload it to the Marmoset server as **exam01**:

    https://cs.ycp.edu/marmoset