CS 201, Fall 2016 — Sep 28th — Exam 1 Name: Solution

Question 1. [5 points] Write code to prompt the user to enter her age, and then based on the age entered, print one of the following messages. If the age is greater than or equal to 18, print "`Hooray! You can vote!`". If the age is less than 18, print"`Too bad, maybe next time`".

```
Scanner keyboard = new Scanner(System.in);
System.out.print("Enter your age: ");
int age = keyboard.nextInt();
if (age >= 18) {
    System.out.println("Hooray! You can vote!");
} else {
    System.out.println("Too bad, maybe next time");
}
```

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 int val;

  public Q2(int v) {
    val = v;
  }
```

```
  public static void main(String[] args) {
    Q2 a = new Q2(4);
    Q2 b = new Q2(5);

    System.out.printf("%d,%d\n",
      a.val, b.val);

    a = b; //make a refer to same object as b
    b.val = 88;
    System.out.printf("%d,%d\n",
      a.val, b.val);
  }
}
```

4 5
88 88

Question 3. [5 points] What output is printed by the following code (which begins on the left and continues on the right)?

```
public class Q3 {
  private static void f(int[] a) {
    a[0] = a[1];
  }
```

a is an alias for the array created in main

```
  public static void main(
      String[] args) {
    int[] p = new int[2];
    p[0] = 17;
    p[1] = 42;
    f(p); // pass reference to the array to f
    System.out.printf("%d,%d\n",
      p[0], p[1]);
  }
}
```

42 42

Question 4. [5 points] Complete the following method so that computes and returns the sum of the elements of the array passed as the parameter.

```
public static double arrSum(double[] arr) {
    double sum = 0.0;
    for (int i = 0; i < arr.length; i++) {
        sum += arr[i];
    }
    return sum;
}
```

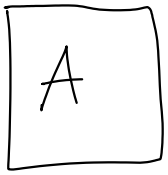
Question 5. [10 points] Consider the following program:

```
public class Q5 {
  public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);
    String fileName = keyboard.nextLine();
    System.out.println("A");
    try {
      FileReader fr = new FileReader(fileName);
      BufferedReader br = new BufferedReader(fr);
      System.out.println("B");
      String line = br.readLine();
      System.out.println("C");
      br.close();
      System.out.println("D");
    } catch (IOException e) {
      System.out.println("E");
    }
  }
}
```

Assume that when the program is run, the user will type the name of a file which may or may not exist.

List all of the possible outputs of the program. (Each will be some combination of "`A`", "`B`", "`C`", "`D`", and "`E`".)

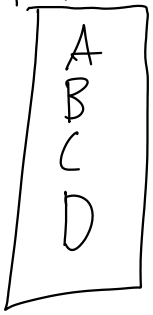
If FileReader constructor throws FileNotFoundException



If call to br.readLine() throws an exception

If call to br.close() throws an exception

If no exception is thrown



Question 6. [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. Assume that `Line` is a class with two `Point` objects representing start and end coordinates, and that `Point` has appropriate constructor and getter methods.

```
public class Line {
  Point start;
  Point end;

  public Line() {
    start = new Point(0,0);
    end   = new Point(0,0);
  }

  private Line(Point start,
               Point end) {
    start = this.start;
    end   = this.end;
  }

  public getStart() {
    return start;
  }

  private Point getEnd(Point end) {
    return end;
  }

  private setStart(Point start) {
    this.start = start;
  }

  public void setEnd() {
    this.end = end;
  }
}
```

should be private

constructors are typically public

reversed

return type missing, should be Point

getters are typically public

getter should not have a parameter

should be public

need return type (void)

missing parameter of type Point

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

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

```
public abstract class Point {
  private   int x;
  private   int y;

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

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

  public int getX() {
    return x;
  }

  public int getY() {
    return y;
  }

  private int X_2() {
    return x*x;
  }

  private int Y_2() {
    return y*y;
  }

  public double distFromOrigin() {
   return sqrt(X_2() + Y_2());
  }

  public abstract double getNormX();

  public abstract double getNormY();
}

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

inherited, but not accessible

private methods are NOT inherited

not inherited (per se) because they are abstract

Vector would need to define concrete implementations of the getNormX and getNormY methods

Programming Questions

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. [30 points] Complete the `Captain` class as follows:

- Declare fields named `firstName` (a String), `lastName` (a String), `gradYear` (an int), and `shipName` (a String).
- Implement 2 constructors - the first accepts and sets first name and last name, and initializes the graduation year to 0, and the ship name to null.
- The second constructor accepts and sets all four class fields (first name, last name, graduation year, and ship name.
- Add getter methods for each of the class fields (`getFirstName`, `getLastName`, etc.)
- Add setter methods ONLY for graduation year and ship name.
- Add a method that returns the Captain's full name in the form "firstName lastName", i.e. "James Kirk". (There should be exactly one space character between the first and last names.)

Hint: Make sure to use method names that match those specified in the JUnit test cases.

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

Question 9. [20 points] Complete the `Target` class as follows:

- Declare `arrows` (an array of `Point`s) that stores the arrow locations that hit this `Target`
- Declare `score` (a double) that stores the calculated score for the arrows for this `Target`
- Define a constructor that takes an `array of Points` parameter, which you should store in `arrows`.
- Define a `calcScore` method that calculates the score for the arrows for this `Target`. You will need to call `calcScore` from the constructor to set the score for this Target. The score for the `Target` is the average distance of all the arrows (`Point`s) in the arrows array from the center of the `Target` (0,0).

Hint: For the `calcScore` method, calculate the distance between each Point and (0,0) in the arrows array, and sum those distances. Then divide by the number of arrows (Points) in the array. Make sure to use method names that match those specified in the JUnit test cases.

Hint: The distance between two points (x_1, y_1) and (x_2, y_2) is $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. In Java you can use the `Math.sqrt` static method to compute the square root of a `double` value.

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

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
```

NOTE: Make sure that you archive the entire project. Also, do not simply accept the default archive file name that shows up in the "To archive file:" drop down box - it will likely be the wrong name.