COS126 Data Types Activity - Section 1.2

Pair Activity: Command-line arguments, Data types, Computation, Type conversion.

1. Write `Eggsactly.java`. Egg cartons each hold exactly 12 eggs. Write a program which reads an integer number of eggs as an argument, then prints out two numbers: how many cartons can be filled by these eggs, and how many eggs will be left over. For example, the output corresponding to

   ```
   java Eggsactly 27 is
   2 3
   ```

   since 27 eggs fill 2 cartons, leaving 3 eggs left over. Hint: use %.

   ```
   // Calculates the number of 12-egg carton can you fill with
   // N eggs, and how many of those N will be left over

   public class Eggsactly {
      public static void main(String[] args) {
         int n = Integer.parseInt(args[0]); // number of eggs

         System.out.print(_____); // number of filled 12-egg cartons
         System.out.print(" ");

         System.out.println(_____); // number of eggs left over
      }
   }
   ```

2. Write `PercentScore.java`.Compute your average score on a two-part exam. You will be given 4 command-line arguments:

   – The number of questions you got right on the first part

   – The total number of questions on the first part

   – The number of questions you got right on the second part

   – The total number of questions on the second part

Output your percentage score on the exam. For example, for `PercentScore 8 10 15 17` since you got a total of 23 questions correct out of 27 and $23/27 = 0.8518$ you should print 85.18518518518519. You may assume the total number of questions is positive.

```java
// Prints your grade based on the number of answers you
// got right on a two-part exam

public class PercentScore {
    public static void main(String[] args) {




    }
}
```

3. Web Exercise 1.2.1. Write `Distance.java`. Given two **integer** command-line arguments, x and y, compute the Euclidean distance of the point (x, y) from the origin $(0, 0)$.

$$distance = \sqrt{x^2 + y^2}$$

Do NOT use `Math.pow(x, 2)` to compute x^2.

```
/*
 *    Compute the distance from (x, y) to the origin.          */
public class Distance {
  public static void main(String[] args) {
    // input point coordinates
    int x = Integer.parseInt




    // compute distance




    // output distance



  }
}
```

4. Exercise 1.2.34. Write `ThreeSort.java`. Given three integer command-line arguments, print them in ascending order. Use `Math.min()` and `Math.max()`.

```
/*
 *    Print the three integer inputs in ascending order.
 *                                                       */
public class ThreeSort {
  public static void main(String[] args) {
    // Input




    // Compute the order




    // Output in ascending order




  }
}
```

- Recommended Exercises: 1.2.4, 1.2.6, 1.2.9, 1.2.13, 1.2.16, 1.2.20 (hint: study Program 1.2.5 on p. 33), 1.2.30