Hershel Thomas
Intro To CS

AddTwo.java

```java
/*
* Adds two given integers and prints the result in a fancy way.
*/
public class AddTwo {
   public static void main(String[] args) {
      int one = Integer.parseInt(args[0]);
      int two = Integer.parseInt(args[1]);
      addTwo(one, two);
   }
   public static void addTwo(int one, int two){
      int answer = one + two;
      System.out.println(one + " + " + two + " = " + answer );
   }
}
```

Coins.java

```java
/*
* Write a program that gets a quantity of cents as a command-line argument.
* The program prints how to represent this quantity using as many quarters as possible,
plus the remainder in cents.
*/
public class Coins {
   public static void main(String[] args) {
      // Put your code here
      int amount = Integer.parseInt(args[0]);
      coins(amount);
   }
   public static void coins(int amount){
      int quarters = amount / 25;
      int cents = amount % 25;
      System.out.printf("Use %d quarters and %d cents\n", quarters, cents);
   }
}
```

GenThree.java

```java
/*
* Generates three random integers, each in a given range [a,b),
* prints them, and then prints the minimal number that was generated.
*/
import java.util.Random;
public class GenThree {
   public static void main(String[] args) {
      // Put your code here
      int minNumber = Integer.parseInt(args[0]);
      int maxNumber = Integer.parseInt(args[1]);
      gen3(minNumber, maxNumber);
   }
   public static void gen3(int min, int max) {

      Random random = new Random();
      int mini = random.ints(min, max) //Using Stream declares random range
            .limit(3)
             //sets the limit of how many random numbers
            .peek(System.out::println)
             //uses .peek to print out each random number on new line
            .min()
            .getAsInt();
           //uses .min function to get the minimum number and sets it to int mini
      System.out.println("The minimal number generated was " + mini);
   }
}
```

LinearEq.java

```
/*
* Solves linear equations of the form a · x + b = c.
* The program gets a, b, and c as command-line arguments,
* computes x, and prints the result.
* Treats the three arguments as well as the computed value as double values
*/
public class LinearEq {
   // Put your code here
   public static void main(String[] args) {
      double a = Double.parseDouble(args[0]);
      double b = Double.parseDouble(args[1]);
      double c = Double.parseDouble(args[2]);
      linearEq(a, b, c);
   }
   public static void linearEq(double a, double b, double c) {
      String equation = String.format("%.1f * x + %.1f = %.1f",a,b,c);
      double x = (c - b)/a;
      System.out.println(equation+"\nx = "+x);
   }
}
```

Triangle.java

```java
/*
* Three sides can form a triangle if the sum of the lengths of any two sides is greater than the length of the remaining side.
* This is known as the Triangle Inequality Theorem.
* Write a program that tests if three given integers form a triangle.
*/
public class Triangle {
  public static void main(String[] args) {
    // Put your code here
    int a = Integer.parseInt(args[0]);
    int b = Integer.parseInt(args[1]);
    int c = Integer.parseInt(args[2]);
    triangle(a, b, c);
  }
  public static void triangle(int a, int b, int c) {
    String nums = String.format("%d, %d, %d: ",a,b,c);
    boolean ans = (a + b) > c;
    System.out.println(nums + ans);
  }
}
```