```
/*
Write a program (AddTwo.java) that adds two given integers and prints the result in a
fancy way.
The command line is: java AddTwo a b.
*/
public class AddTwo {
    public static void main(String[] args) {

        // Declare variables to store the given integers
        int num1 = Integer.parseInt(args[0]);
        int num2 = Integer.parseInt(args[1]);

        // Calculate the sum of the integers given
        int sum = num1 + num2;

        // Print the sum pf the integers
        String output = String.format("%d + %d = %d", num1, num2, sum);
        System.out.println(output);
    }

}
```

```
/*
Assume that there are two coins only: A coin of 25 cents, called quarter, and a coin of a single
cent, called cent. Write a program (Coins.java) that gets a quantity of cents as a command-line
argument, and 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) {

        // Declare a variable to store the number of cents received
        int quantity = Integer.parseInt(args[0]);

        // Calculate the number of quarters and cents and declare a variable to store each of them
        int quarters = quantity / 25;
        int cents = quantity % 25;

        // Print the number of quarters and cents required
        String output = String.format("Use %d quarters and %d cents", quarters, cents);
        System.out.println(output);
    }

}
```

```
/*
Write a program (LinearEq.java) that 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. Assume
that a is not zero. The program treats the three arguments as well as the computed value as
double values. The program prints the equation, and its solution.
 */
public class LinearEq {
    public static void main(String[] args) {

        // Declare variables to store the equation's variables
        double a, b, c;
        a = Double.parseDouble(args[0]);
        b = Double.parseDouble(args[1]);
        c = Double.parseDouble(args[2]);

        // Solve the linear equation and store the result in a variable
        double x = (c - b) / a;

        // Print the solution of the equation
        System.out.println(a + " * x + " + b + " = " + c + "\nx = " + x);
    }
}
```

```
/*
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. For
example, the
three numbers 3, 4, 5 form a triangle, and the three numbers 2, 3, 6 don’t form a
triangle. Write
a program (Triangle.java) that tests if three given integers form a triangle.
 */
public class Triangle {
   public static void main(String[] args) {

      // Declare variables to store the size of the three sides of the suspected triangle
      int side1, side2, side3;
      side1 = Integer.parseInt(args[0]);
      side2 = Integer.parseInt(args[1]);
      side3 = Integer.parseInt(args[2]);

      // Declare a boolean variable to store the answer: is it a triangle?
      boolean isTriangle = side1 + side2 > side3 &&
                           side1 + side3 > side2 &&
                           side2 + side3 > side1;

      // Print the size of the sides and the answer
      String output = String.format("%d, %d, %d: %b", side1, side2, side3, isTriangle);
      System.out.println(output);
   }
}
```

```
/*
Write a program (Gen3.java) that generates three random integers, each in a given
range [a,b),
i.e. greater than or equal to a and less than b, prints them, and then prints the
minimal number
that was generated
 */
public class GenThree {
    public static void main(String[] args) {
        // Declare variables to store upper and lower bounds, and the range between
them
        int upperBound, lowerBound, range;
        lowerBound = Integer.parseInt(args[0]);
        upperBound = Integer.parseInt(args[1]);
        range = upperBound - lowerBound;

        double num1, num2, num3;

        // Generate three random numbers between 0.0 (inclusive) and 1.0 (exclusive)
        num1 = Math.random();
        num2 = Math.random();
        num3 = Math.random();

        // Scale the random numbers to fit within the specified range
        num1 = num1 * range;
        num2 = num2 * range;
        num3 = num3 * range;

        // Declare three integers that represents the generated random numbers in the
given range
        int gen1 = (int) (num1 + lowerBound);
        int gen2 = (int) (num2 + lowerBound);
        int gen3 = (int) (num3 + lowerBound);

        // Find the minimum generated number among gen1, gen2, and gen3
        int minNum = (int) (Math.min(Math.min(gen1, gen2), gen3));

        // Print the generated numbers and the minimum number among them
        String output = String.format(
                "%d\n%d\n%d\nThe minimal generated number was %d", gen1, gen2,
gen3, minNum);
        System.out.println(output);
    }
}
```