

Taking our Student and Classroom example from earlier, you should fill in the method `getMostImprovedStudent`, as well as the method `getExamRange`. The most improved student is the one with the largest exam score range.

To compute the exam score range, you must subtract the minimum exam score from the maximum exam score.

For example, if the exam scores were 90, 75, and 84, the range would be $90 - 75 = 15$.

```
public class Classroom
{
    Student[] students;
    int numStudentsAdded;

    public Classroom(int numStudents)
    {
        students = new Student[numStudents];
        numStudentsAdded = 0;
    }

    public Student getMostImprovedStudent()
    {
        // Fill in this method.
    }

    public void addStudent(Student s)
    {
        students[numStudentsAdded] = s;
        numStudentsAdded++;
    }

    public void printStudents()
    {
        for(int i = 0; i < numStudentsAdded; i++)
        {
            System.out.println(students[i]);
        }
    }
}
```

```
import java.util.*;

public class Randomizer{

    public static Random theInstance = null;

    public Randomizer(){

    }

    public static Random getInstance(){
        if(theInstance == null){
            theInstance = new Random();
        }
        return theInstance;
    }

    public static boolean nextBoolean(){
        return Randomizer.getInstance().nextBoolean();
    }

    public static boolean nextBoolean(double probability){
        return Randomizer.nextDouble() < probability;
    }

    public static int nextInt(){
        return Randomizer.getInstance().nextInt();
    }

    public static int nextInt(int n){
        return Randomizer.getInstance().nextInt(n);
    }

    /* Return a number between min and max, inclusive. */
    public static int nextInt(int min, int max){
        return min + Randomizer.nextInt(max - min + 1);
    }

    public static double nextDouble(){
        return Randomizer.getInstance().nextDouble();
    }

    public static double nextDouble(double min, double max){
```

```
        return min + (max - min) * Randomizer.nextDouble();
    }
}
```

```
public class ClassroomTester extends ConsoleProgram
{
    public void run()
    {
        Classroom c = new Classroom(2);

        Student ada = new Student("Ada", "Lovelace", 12);
        ada.addExamScore(44);
        ada.addExamScore(65);
        ada.addExamScore(77);

        Student alan = new Student("Alan", "Turing", 11);
        alan.addExamScore(38);
        alan.addExamScore(24);
        alan.addExamScore(31);

        // add students to classroom
        c.addStudent(ada);
        c.addStudent(alan);
        c.printStudents();

        Student mostImproved = c.getMostImprovedStudent();
        System.out.println("The most improved student is " + mostImproved.getName());
    }
}
```

```

public class Student
{
    private static final int NUM_EXAMS = 4;

    private String firstName;
    private String lastName;
    private int gradeLevel;
    private double gpa;

    private int[] exams;
    private int numExamsTaken;

    /**
     * This is a constructor. A constructor is a method
     * that creates an object -- it creates an instance
     * of the class. What that means is it takes the input
     * parameters and sets the instance variables (or fields)
     * to the proper values.
     *
     * Check out StudentTester.java for an example of how to use
     * this constructor.
     */
    public Student(String fName, String lName, int grade)
    {
        firstName = fName;
        lastName = lName;
        gradeLevel = grade;
        exams = new int[NUM_EXAMS];
        numExamsTaken = 0;
    }

    public int getExamRange()
    {
        // Fill in this method.
    }

    public String getName()
    {
        return firstName + " " + lastName;
    }

    public void addExamScore(int score)
    {

```

```
    exams[numExamsTaken] = score;
    numExamsTaken++;
}

// This is a setter method to set the GPA for the Student.
public void setGPA(double theGPA)
{
    gpa = theGPA;
}

/**
 * This is a toString for the Student class. It returns a String
 * representation of the object, which includes the fields
 * in that object.
 */
public String toString()
{
    return firstName + " " + lastName + " is in grade: " + gradeLevel;
}
}
```