Lab 13

1. Create an Integer ArrayList in the main method.

```java
import java.util.ArrayList;

public class Lab13 {

    public static void main(String[] args) {
        ArrayList<Integer> numbers = new ArrayList<>();
    }
}
```

2. Create a method that receives an integer that represents the size of an Integer ArrayList. The method should create an Integer ArrayList of the given size and fill it with random integers between 1 and 100. The method should return the ArrayList.

```java
    public static ArrayList<Integer> fillArrayList(int size) {
        ArrayList<Integer> numbers = new ArrayList<>();
        for (int i = 0; i < size; i++) {
            numbers.add((int) (Math.random() * 100) + 1);
        }
        return numbers;
    }
```

3. Create a method that that removes any value from the ArrayList that is less than 50. The method should return the ArrayList.

Hint: Structure your loop correctly so elements in the ArrayList are not skipped.

```java
    public static ArrayList<Integer> removeLessThan50(ArrayList<Integer>
numbers) {
        for (int i = 0; i < numbers.size(); i++) {
            if (numbers.get(i) < 50) {
                numbers.remove(i);
                i--;
            }
        }
        return numbers;
    }
```

4. Call the previous methods (2,3) and print out the ArrayList before and after the values less than 50 are removed.

5. Create a program the prompts the user to enter times for a runner. The values should be saved as Doubles. You should have methods that find the fastest and slowest times. The program should print out the fastest and slowest times.

```java
import java.util.*;

public class RunnerTimes {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Double> times = new ArrayList<>();

        while(true) {
            System.out.println("Enter time for the runner (0 to exit): ");
            Double time = scanner.nextDouble();
            if (time == 0) {
                break;
            }
            times.add(time);
        }

        if (times.size() > 0) {
            Double fastestTime = findFastestTime(times);
            Double slowestTime = findSlowestTime(times);

            System.out.println("Fastest time: " + fastestTime);
            System.out.println("Slowest time: " + slowestTime);
        } else {
            System.out.println("No times entered.");
        }
    }

    public static Double findFastestTime(ArrayList<Double> times) {
        Double fastest = times.get(0);
        for (int i = 1; i < times.size(); i++) {
            if (times.get(i) < fastest) {
                fastest = times.get(i);
            }
        }
        return fastest;
    }

    public static Double findSlowestTime(ArrayList<Double> times) {
        Double slowest = times.get(0);
        for (int i = 1; i < times.size(); i++) {
            if (times.get(i) > slowest) {
                slowest = times.get(i);
            }
        }
        return slowest;
    }
```

~~~
6. Add another ArrayList that saves the names of the runners. The program
should print out the name of the runner with the fastest and slowest
times.

**Hint:** Use the index to match the times with the names.

```java

import java.util.*;

public class RunnerTimes {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Double> times = new ArrayList<>();
        ArrayList<String> names = new ArrayList<>();

        while(true) {
            System.out.println("Enter name of the runner (0 to exit): ");
            String name = scanner.nextLine();
            if (name.equals("0")) {
                break;
            }
            names.add(name);

            System.out.println("Enter time for the runner: ");
            Double time = scanner.nextDouble();
            times.add(time);
        }

        if (times.size() > 0) {
            Double fastestTime = findFastestTime(times);
            Double slowestTime = findSlowestTime(times);

            int fastestIndex = times.indexOf(fastestTime);
            int slowestIndex = times.indexOf(slowestTime);

            System.out.println("Fastest time: " + fastestTime + " by " +
names.get(fastestIndex));
            System.out.println("Slowest time: " + slowestTime + " by " +
names.get(slowestIndex));
        } else {
            System.out.println("No times entered.");
        }
    }

    public static Double findFastestTime(ArrayList<Double> times) {
        Double fastest = times.get(0);
        for (int i = 1; i < times.size(); i++) {
            if (times.get(i) < fastest) {
                fastest = times.get(i);
~~~

```
            }
        }
        return fastest;
    }

    public static Double findSlowestTime(ArrayList<Double> times) {
        Double slowest = times.get(0);
        for (int i = 1; i < times.size(); i++) {
            if (times.get(i) > slowest) {
                slowest = times.get(i);
            }
        }
        return slowest;
    }
}
```