



---

ภาษาการเขียนโปรแกรม

WEEK 6

02-212-213

อ.ธิดาวรรณ คล้ายศรี

---



## Fundamental of Computer Programming

- Review Data Structure
- Review Basic/essential Algorithms

- 1.1 An Introduction to Programming (บทนาการเขียนโปรแกรม)
- 1.2 Traditional Programming Languages (ภาษาการเขียนโปรแกรมแบบดั้งเดิม)
- 1.3 Syntax of the Programming Languages (ไวยากรณ์ของภาษาเขียนโปรแกรม)
  - 1.3.1 Control Block (การควบคุมบล็อก)
  - 1.3.2 Block Structure (โครงสร้างบล็อก)
  - 1.3.3 Recursive (การทำงานแบบเรียกตัวเอง)
- 1.4 Review Data Structure (ทบทวนโครงสร้างข้อมูล)
- 1.5 Sorting and Searching Algorithms (อัลกอริทึมจัดลำดับและสืบค้นข้อมูล)
- 1.6 Testing Programs (การทดสอบโค้ด)
  - 1.6.1 Compiler
  - 1.6.2 Running Time and Debugging
  - 1.6.3 Creating Test Cases



## 1.6 TESTING PROGRAMS

ทดสอบโปรแกรม

# Creating Test Case

ทำการทดสอบผลลัพธ์การจัดลำดับข้อมูลจากโปรแกรมที่เขียน กับผลลัพธ์ที่คาดหวัง

---

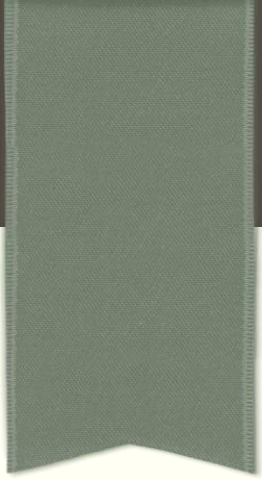
ผลลัพธ์

Scores
65
72
83
85
71
60
55

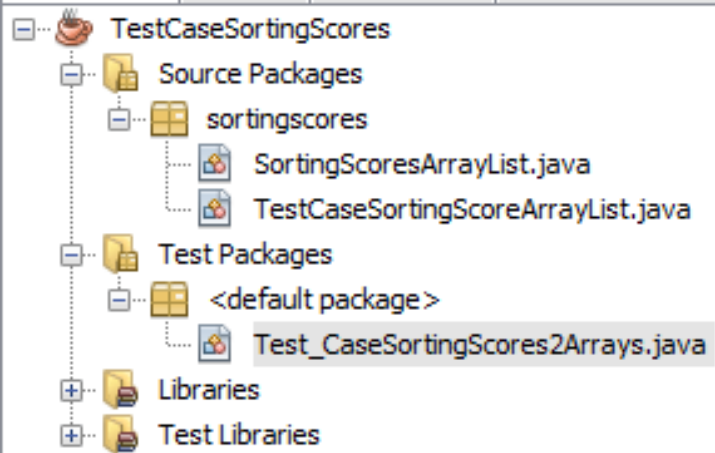
ผลลัพธ์ที่คาดหวัง

Answer
65
72
83
85
71
60
55

เทียบกับ



Laboratory



```
16 public class Test_CaseSortingScores2Arrays {
17
18     public Test_CaseSortingScores2Arrays () {
19     }
20
21
22     @Test
23     public void testCase() throws FileNotFoundException {
24         SortingScoresArrayList S = new SortingScoresArrayList();
25         TestCaseSortingScoreArrayList T = new TestCaseSortingScoreArrayList();
26
27         T.get2ArrayLists(S, "AnswerScores.txt");
28         System.out.println(T.comparing2ArrayLists());
29     }
30 }
31
```

```
public void get2ArrayLists(SortingScoresArrayList S, String fName) {  
    S.readScoreFromTextFile("Scores.txt"); //1) Reading data score from a text file into ArrayList/Collection  
    System.out.println("Original Scores:");  
    S.printScores();//display the array list of scored  
    S.sortScoresAtoZ();//2) Sorting data using Collection.sort (small to large)  
    System.out.println("\nSorted Scores:");  
    S.printScores();//3) Displaying the sorted data  
  
    /*Reading expected output*/  
    score= S.getScores();  
    Scanner fln = new Scanner(new FileReader(fName));//reads students' scores from a text file.  
    do{  
        ans.add(fln.nextInt());  
    }while(fln.hasNextInt());  
  
}
```

```
public Boolean comparing2ArrayLists(){
```

---

```
    Boolean test = true;
    System.out.println("\nComparison of the Two ArrayLists");
    System.out.println("-----");
    System.out.println("Indexes\tScores\tExpectedScores");
    System.out.println("-----");
    for(int i=0; i<score.size(); i++){
        if(Objects.equals(score.get(i), ans.get(i))){
            System.out.println "[" + i + "] " + score.get(i) + " = " + ans.get(i);
        } else {
            System.out.println "[" + i + "] " + score.get(i) + " != " + ans.get(i);
            test = false;
        }
    }
    return test;
}
```





งานเดี่ยวปฏิบัติเขียนโปรแกรม

## ■ เขียนโปรแกรมตัดเกรด พร้อมทั้งเขียนโค้ดเพื่อทดสอบความถูกต้อง

---

### ■ นั่งประจำเครื่องตามกลุ่มเดิม

1) อ่านข้อมูลชื่อนศ.(name) และคะแนนรวม (score) นศ. จำนวน ... คน (N) ของ ... 58/... มาจาก **Text file**

2) ให้ทำการตัดเกรดแบบอิงเกณฑ์ F, D, D+, ... A;

■  $\text{score} < 50 \rightarrow \text{grade F}$ ,  $50 \leq \text{score} < 55 \rightarrow \text{grade D}$ ,

$55 \leq \text{score} < 60 \rightarrow \text{grade D+}$ , ... ,  $80 \leq \text{score} \rightarrow \text{grade A}$ ,

3) โปรแกรมต้องสามารถ

**3.1 แสดงผลเกรดนศ. โดยให้เรียงตามคะแนน**

**3.2 สามารถทำการสืบค้นชื่อเพื่อสอบถามคะแนนและเกรดได้**

## ทำ Lab Testing ผลลัพธ์การจัดเรียงข้อมูล

---

- Read text file ข้อมูล score ที่ยังไม่ได้เรียงมาจาก Text file
- Sorting score
- Read text file score ที่ว่าจะได้ผลลัพธ์
- เปรียบเทียบ 2 Arrays/ArrayLists