CONFIDENTIAL

UNIVERSITI TEKNOLOGI MALAYSIA
FACULTY OF COMPUTING

SKILL-BASED TEST 2

SEMESTER I 2017/2018

SUBJECT CODE	: SCSJ 1013
SUBJECT NAME	: PROGRAMMING TECHNIQUE I
YEAR/COURSE	: 1 (SCSJ/SCSV/SCSB/SCSR/SCSP)
TIME	: 4:30 PM – 5.45 PM
DATE	: 5 DECEMBER 2017
VENUE	: N28 MPK1-MPK10

INSTRUCTIONS TO THE STUDENTS:

- This test consists of only **ONE** question.
- References to any resources by any means are strictly prohibited.
- You are given **ONE HOUR FIFTEEN MINUTES** to complete the test inclusive the submission of your program.

MATERIAL FOR THE TEST:

- You are provided a program source code file named **`SBT2_051217.cpp`** which you want to use it as the base to answer this test, and sample of input file named **`scores.txt`**. All these files are compressed into a RAR file.
- Download the RAR file from the UTM's elearning to your computer's hard drive.
- **IMPORTANT NOTES:** Do not edit the source code file directly from the RAR file. You must extract them into your local hard drive.

SUBMISSION PROCEDURE:

- Only the source code (i.e. the edited **`SBT2_051217.cpp`**) is required for the submission.
- Submit the source code via the UTM's e-learning system.

This question booklet consists of 4 pages inclusive of the cover page.

Problem 60 Marks

Johor Darul Ta'zim Football Club, or simply known as JDT is the most popular football team in Malaysia. JDT is the first team in the country to win four consecutive league titles (2014, 2015, 2016, and 2017). Before this achievement, no team had ever won the league championship more than twice in a row since the league system was first introduced in Malaysia in 1979.

Assumed that you are a UTM practical student currently attached to this club. You have been asked to write a program for your supervisor to facilitate the JDT Manager to keep track the performance scores of each player.

There are seven tasks already listed to complete the program. Edit the source code **`SBT2_051217.cpp`** according to the tasks given by the supervisor. The tasks given as follows;

TASK 1:
Write a function named **`readScores`** that takes two array parameters, an array of strings and an array of integers. The function should scan through the input file named **`scores.txt`**. Make sure you only continue reading the file if the file is successfully opened, otherwise print the error message and exit the program. Set the string array entry at index 0 to the name of player along with the scores of the respective player. [12 Marks]

TASK 2:
Complete the definition of function **`getHighestScore`**. This function should compare the scores among the players and then find the highest score. Once identified, this function returns the index of array with the highest score. [6 Marks]

TASK 3:
Complete the definition of function **`getLowestScore`**. This function works similar with TASK 2 but it's for finding the lowest score. [6 Marks]

TASK 4:

Complete the definition of function `averageScore`. This function calculates the average of the scores and return this value back into calling function. [5 Marks]

TASK 5:

Using appropriate functions defined above, read inputs from the input file, then find the highest and lowest score. Finally, calculate the average of the scores. [9.5 marks]

TASK 6:

Continue to print all the results from Task 5 into the file named **`report.txt.`** ***Note:*** Use proper output formatting. Your full report should look like this:

```
PLAYER'S NAME  SCORES
=============  ======
Farizal         7852
Hasbullah       6643
Afiq            6561
Marcos          7332
Safiq           8991
Hazwan          7013
Gabriel         7116
Kunanlan        6986
Darren          6559
Safawi          6890

HIGHEST SCORE : 8991 (Safiq)

LOWEST SCORE  : 6559 (Darren)

AVERAGE SCORE : 7194.30
```

[18.5 Marks]

TASK 7:

The program is able to run and display correct output in the output file. [3 Marks]

The assessment criteria is shown in **Table 1**.

Table 1: Assessment Criteria

Task	Criteria	Marks
1	Declare input file object in function `readScores`	0.5
	Open and close input file	2.5
	Check if the file is successfully open ⇨ if fail, display error message and exit the program	2 2
	Get the names and scores from input file and store them in two parallel arrays – string and integer	5
2	Declare variable in function `getHighestScore`	1
	Find the index highest score	4.5
	Return the index highest score	0.5
3	Declare variable in function `getLowestScore`	1
	Find the index lowest score	4.5
	Return the index lowest score	0.5
4	Declare variable in function `averageScore`	1
	Find the average score	3.5
	Return the average score	0.5
5	Call `readScores` function	2
	Call `getHighestScore` function	2.5
	Call `getLowestScore` function	2.5
	Call `averageScore` function	2.5
6	Declare output file object in function `readScores`	0.5
	Open and close output file	2.5
	Print all player’s name and scores, highest and lowest score with player’s name, and average of scores in output file with proper output formatting	15.5
7	The program is able to run	3
Total		**60**