

TUTORIAL 6 (SECJ1013)
PROGRAMMING TECHNIQUE 1
SECTION 04 & 07, SEM 1, 2024/2025

Q#1

Given an incomplete program below, complete the program by filling in the blanks with an appropriate C++ statement according to the instructions stated in the comments. This program intends to read the data from the input file (“**fileInp.txt**”) and to display the data into the output file (“**fileOut.txt**”). The data in the input file is shown in **Figure 1** and the expected data in the output file is shown in **Figure 2**.

```
1  #include <iostream>
2  #include <fstream>
3  #include <iomanip>
4  using namespace std;
5
6  int main()
7  {
8      float num, average = 0, sum = 0;
9      int cnt = 0;
10     fstream inputfile, outputfile;
11
12     //a) Open the input file named "fileInp.txt"
13     _____
14
15     //b) Open the output file named "fileOut.txt"
16     _____
17
18     //c) Check for successful opening file
19     _____
20     cout << "ERROR!! Input file could not be opened\n";
21     else {
22         //d) Read data from the input file
23         while _____ {
24
25             //e) Write data to the output file
26             _____
27
28             //f) Calculate sum of all data
29             _____
30             cnt++;
31         }
32
33         //g) Calculate average
34         _____
35
36         //h) Display average to the output file
37         _____
38
39         //i) Close the input file
40         _____
41
42         //j) Close the output file
43         _____
```

44	
45	return 0;
46	}

1	2	3	.	4	5	6			
0	.	0	0	0	0	3	4	5	6
5	.	6	7	8	9				

Figure 1: Input file

							1	2	3						
	3	.	4	6	e	-	0	0	5						
						5	.	6	8						
a	v	e	r	a	g	e	=	4	3	.	0	4	5		

Figure 2: Output file

Q#2

Write the output of the following program segment in **Table 3**. Use # to represent whitespace.

1	int num1 = 222, num2 = 12354;
2	double n1 = 123.4567, n2 = 0.000001234;
3	cout << right << " " << setw(6) << num1 << " " << endl;
4	cout << left << " " << setw(4) << num2 << " " << endl;
5	cout << showpoint;
6	cout << n1 << " " << n2 << endl;
7	cout << setprecision(2);
8	cout << n1 << " " << n2 << endl;
9	cout << fixed;
10	cout << n1 << " " << n2 << endl;

Table 3

Line	Output
3	
4	
6	
8	
10	

Q#3

- a) Based on the outputs given in **Figure 1**, complete the **Program 1** with appropriate C++ output formatting, i.e. **setw**, **setprecision**, **showpoint**, and **fixed**. *Note*: one cell represents one space per digit.

Line	Outputs
10	1 0 4 5 . 2 3 3 3 . 4 4 0 0
11	1 . 3
12	1 . 0 5 e + 0 0 3
14	1 . 2 5 6
15	3 3 . 4 4 0
16	1 0 4 5 . 2

Figure 1: Outputs for Program 1

```

1 //Program 1
2 #include <iostream>
3 #include <iomanip>
4 using namespace std;
5
6 int main()
7 {
8     float a = 1045.234, b = 1.25551, c = 33.44;
9
10    cout << _____ (i) _____ << a << " " << c << endl;
11    cout << _____ (ii) _____ << b << endl;
12    cout << _____ (iii) _____ << a << endl;
13
14    cout << _____ (iv) _____ << _____ (v) _____ << b << endl;
15    cout << _____ (vi) _____ << c << endl;
16    cout << _____ (vii) _____ << a << endl;
17
18    return 0;
19 }

```

- b) Based on the **Program 2** and inputs given in **Table 1**, write the output to be displayed for each of the corresponding codes in **Code 1**, **Code 2**, and **Code 3**.

```

1 //Program 2
2 #include <iostream>
3 #include <iomanip>
4 using namespace std;
5
6 int main()
7 {
8     char w[20];
9
10    cout << "Enter a string: ";
11    // either Code 1, Code 2 or Code 3
12    // will be placed here
13
14    return 0;
15 }

```

Table 1: Codes and inputs for Program 2

Codes	Inputs
//Code 1 cin.getline(w, 10); cout << w << endl;	Final Exam PT1
//Code 2 cin >> setw(9) >> w; cout << w << endl;	Good Luck!

//Code 3

```
cin >> setw(5) >> w;  
cout << w << endl;
```

Programming

Given a C++ program as shown in Figure 1. Answer the question follows:

```
1  #include <iostream>  
2  #include <cmath>  
3  #include <iomanip>  
4  
5  using namespace std;  
6  
7  void findMe(double a, double b);  
8  
9  int main()  
10 {   double first, second;  
11  
12     cout << "Enter two numbers :";  
13     cin >> first >> second;  
14     cout << endl;  
15  
16     findMe(first, second);  
17  
18     return 0;  
19 }  
20  
21 void findMe(double a, double b)  
22 {   double c;  
23  
24     if (a != 0)  
25         c = sqrt(b)/a;  
26     else{  
27         cout << "Enter a nonzero number: ";  
28         cin >> a;  
29         cout << endl;  
30         c = pow(b,a);  
31     }  
32  
33     cout << fixed << showpoint << setprecision(2);  
34     cout << a << ", " << b << ", " << c << endl;  
35 }
```

Figure 1

Q#4

Program 2 is intended to read the data from the input file ("*input.txt*") and to display the data on the screen. The data in the input file is shown in **Figure 1**. Fill in the blank with appropriate C++ statements.

```
2  10  14  88  77  20  101  -1  40  -57
```

Figure 1: Input file

Line	Program 2
1	#include <iostream>
2	_____ //Include suitable library for file
3	_____ //operations
4	using namespace std;
5	
6	int main()
7	{
8	int i, num;
9	fstream inDat;
10	
11	//Open the input file named "input.txt"
12	_____;
13	//Test for successful opening file
14	_____;
15	{
16	cout << "ERROR cannot open the file!" << endl;
17	return 0;
18	}
19	cout << "The numbers are : \n";
20	for (i = 1; i <= 10; i++)
21	{
22	//Read the numbers from the input file
23	_____;
24	
25	//Display the numbers on the screen
26	_____;
27	}
28	cout << "Exit from the program";
29	
30	//Close the input file
31	_____;
32	return 0;
33	}
34	

Q#5

- a. Based on the **Program 4**, demonstrate the outputs in **Table 4**. *Note:* one cell represents one space per digit or character.

Line	Program 4
1	#include <iostream>
2	#include <iomanip>
3	using namespace std;
4	
5	int main()
6	{

7	float a = 89.74, b = 6.7893412, c = 33;
8	
9	cout << left << setprecision(3);
10	cout << setw(5) << a << setw(10) << b << endl;
11	
12	cout << right << fixed << setprecision(4);
13	cout << setw(7) << a << setw(8) << b << endl;
14	return 0;
15	}

Answer:

Table 4: Outputs for Program 4

Line	Outputs															
12																
15																

- b. Based on the **Program 5** and using the input given in **Table 5**, demonstrate the corresponding output to be displayed for each of the corresponding codes in Code 1 and Code 2. Write the output in **Table 6**.

1	// Program 5
2	#include <iostream>
3	#include <iomanip>
4	
5	int main()
6	{
7	char word[20], word2[5];
8	
9	cout << "Enter a string: ";
10	// either Code 1 or 2
11	// will be placed here
12	
13	return 0;
14	}

Table 5: Codes and inputs for Program 3

Codes	Inputs
//Code 1 cin.getline(word, 8); cout << setw(15) << word << endl;	How are you? Good luck!
//Code 2 cin >> setw(10) >> word; cout << setw(15) << word << endl;	HowAreYou?GoodLuck!

Answer:

Table 6: Output for Program 5

Code	Output															
1																
2																

Q#6

Trace the executions of the program with the following set of inputs:

Set input #1: 2 16

Set input #2: 8 64

Set input #3: 0 25 1

Note: provide the trace table (as shown below) in your answers

Trace table:

Set input #	first	second	a	b	c	Output
1						
2						
3						

Q#7

The 9-Eleven Mart is a convenience store company operating in the state of Johor. The company has five stores located in several branches; Johor Bahru, Segamat, Batu Pahat, Kota Tinggi and Mersing, respectively. At the end of each year, the management of the company wants to know the performance of their company. They have decided to use a computer program to help them in analyzing the company's sales. You, as a freelance programmer have been appointed to develop the program using C++ language. The requirements of the program are as follow:

Input:

- The program should read in sales data from a text file.
- The name of the input file has to be entered by the user.
- The format of the input file is as follows: The first to twelfth columns indicate the sales for each month, i.e., the first column is for the sales of January, second column is for February, third column is for March, and so forth. The last column indicates store branches. Note that sales in each cell is represented in multiple of RM 1000.00
- Figure 2 shows an example of input file named "sales2012.txt" containing sales data for the year 2012.

94	49	96	67	82	34	91	64	15	97	98	78	Johor Bahru
71	57	17	31	63	38	77	74	61	22	27	59	Segamat
36	16	30	19	29	41	23	25	22	37	28	29	Batu Pahat
87	48	49	91	72	69	13	97	43	41	29	58	Kota Tinggi
34	32	74	57	32	80	76	40	64	48	41	68	Mersing

Figure 2: An example of input file, “sales2012.txt”

Output:

- The program should print out a report into a text file.
- The name of the output file has to be entered by the user.
- The report should include:
 - The grand total of sales, i.e., over all stores throughout the year.
 - The average sales per month.
 - The highest sales. Print the store, month and the sales whose the highest sales.
 - The total sales for each month. The months should be printed with their abbreviated names, such as, “Jan”, “Feb”, “Mar”, and so forth.
 - The total sales for each store.
 - The list of profitable stores. A store is considered profitable if it manages to achieve minimum annual sales of RM 600,000.00.
- Note that all money values have to be specified as with 2 decimal points, 10 spaces in width, and right-justified.
- Figure 3 shows an example report file for the sales data of the year 2012.

Arrays:

- The program should use a two-dimensional array to store the sales data.
- The program should also use a one- dimensional array to store the store branches.

Functions:

The program should have the following function:

- **readFile.** The purpose of this function is to read the sales data and store branches from an input file , then put the read data into a two-dimensional array (for the sales data), and a one-dimensional array (for the store branches). The file’s name and both the arrays have to be accepted as arguments for the function.
- **grandTotalSales.** This function should return the grand total of sales over all stores throughout the year. It should accept a two-dimensional array (representing sales) as its arguments.

- **averageMonthlySales.** This function should return the company's average sales per month. It should accept a two-dimensional array (representing sales) as its arguments.
- **monthTotalSales.** This function should accept a two-dimensional array and the index of a column in the array (representing a month) as its arguments. The function should return the total sales for the specified month.
- **storeTotalSales.** This function should accept a two-dimensional array and the index of a row in the array (representing a store branch) as its arguments. The function should return the total sales for the specified branch.
- **indicesOfHighestSales.** This function should accept a two-dimensional array (representing sales) as one of its arguments. It should determine the indices of row and column of a cell in the array whose the highest sales.

Write a complete C++ program based on the requirements given above.

```

Grand total of sales over all stores: RM 3140000.00
Average sales per month:           RM  261666.67

The highest sales:
-----
Store: Johor Bahru
Month: Nov
Sales: RM 98000.00

Total sales by month:
Month  Sales
-----
Jan    RM322000.00
Feb    RM202000.00
Mar    RM266000.00
Apr    RM265000.00
May    RM278000.00
Jun    RM262000.00
Jul    RM280000.00
Aug    RM300000.00
Sep    RM205000.00
Oct    RM245000.00
Nov    RM223000.00
Dec    RM292000.00

Total sales by store:
Store      Total Sales
-----
Johor Bahru  RM 865000.00
Segamat     RM 597000.00
Batu Pahat  RM 335000.00
Kota Tinggi RM 697000.00
Mersing     RM 646000.00

Profitable stores:
-----
Johor Bahru
Kota Tinggi
Mersing

```

Figure 3: The output file for the sales data in Figure 2

Q#8

A tele match event has been held in Sekolah Rendah Tebing Tinggi. Three teams are allowed to participate in this match, with each team consisting of four participants. Five (5) events were contested, namely E1, E2, E3, E4 and E5. Table 1 shows the scores that have been collected by each team for the five events.

INSTRUCTIONS:

Write a C++ program which can assist the tele match committee to determine the winner for these events. Your program should be able to do the following tasks:

- (a) The program will read input data: team ID, participant ID and **scores** for the five events namely E1, E2, E3, E4 and E5 from an input file named "**input.txt**" into an array **marks[12][7]** of type **int**. Example of the series of input data in input file is shown in Figure 4.

Table 1: Collected scores

Team ID	Participant ID	E1	E2	E3	E4	E5
1	1001	10	5	8	10	6
	1002	8	7	10	7	9
	1003	7	10	10	6	10
	1004	10	10	8	7	7
2	2001	7	8	10	9	10
	2002	10	8	7	8	10
	2003	8	6	8	8	10
	2004	7	8	8	8	8
3	3001	10	9	10	10	10
	3002	8	7	8	8	8
	3003	7	8	9	10	6
	3004	8	6	8	7	7

- (b) The program must be able to notify the user if the input file cannot be opened (failed to open) with proper prompt. The example for user notification where the file fails to open is shown in Figure 5.

```
1 1001 10 5 8 10 6
1 1002 8 7 10 7 9
1 1003 7 10 10 6 10
1 1004 10 10 8 7 7
2 2001 7 8 10 9 10
2 2002 10 8 7 8 10
2 2003 8 6 8 8 10
2 2004 7 8 8 8 8
3 3001 10 9 10 10 10
3 3002 8 7 8 8 8
3 3003 7 8 9 10 6
3 3004 8 6 8 7
```

Figure 4: Input file named "**input.txt**"

```
Sorry, input file does not exist!
```

Figure 5: Example user notification in case file fails to open

- (c) Calculate the total score for each participant.
 (d) Calculate the total score for each team.

- (e) Besides the function `main()`, the program needs to define **three** (3) other functions as described in Table 2. Use appropriate argument (if necessary) for each function.
- (f) The program needs to print out the following information. Figure 6 shows the example, run of the successful program.
- The team ID.
 - The participant ID.
 - The **scores** for the five events, E1, E2, E3, E4 and E5 for each participant.
 - The total score for each participant.
 - The total score for each team.
 - The winner for individual category (selected based on highest total score that collected by the participants).
 - The winner for group category (selected based on highest total score that collected by the teams).

Id	E1	E2	E3	E4	E5	Total
TEAM 1						
1001	10	5	8	10	6	39
1002	8	7	10	7	9	41
1003	7	10	10	6	10	43
1004	10	10	8	7	7	42
TOTAL						165
TEAM 2						
2001	7	8	10	9	10	44
2002	10	8	7	8	10	43
2003	8	6	8	8	10	40
2004	7	8	8	8	8	39
TOTAL						166
TEAM 3						
3001	10	9	10	10	10	49
3002	8	7	8	8	8	39
3003	7	8	9	10	6	40
3004	8	6	8	7	7	36
TOTAL						164
Winner for Individual Category: 3001 (Team 3)						
Winner for Group Category: Team 2 (Score = 166)						
Press any key to continue . . .						

Figure 6: Output of the program

Table 2: Description for functions

Function	Description
displayLine()	To display lines using the 52 characters of '-'. The function should use loop to display the line.
findIndWinner()	To determine the winner for individual category (selected based on the highest total score that was collected by participants). The function should accept the array for a total score of each participant as one of its argument.
findTeamWinner()	To determine the winner of group category (selected based on the highest total score that was collected by teams). The function should accept the array for a total score for each team as one of its argument.