



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SCHOOL OF COMPUTING
Faculty of Engineering

Semester I 2020/2021

SECJ1013-05 Programming Technique I

Assignment 4

Lecturer's Name: Dr. Goh Eg Su

Date Submitted: 2 February 2021

Group Members:

Name	Matric No.
Nurarissa Dayana Binti Mohd Sukri	A20EC0120
Madiah binti Che Zabri	A20EC0074
Sakinah Al'izzah Binti Mohd Asri	A20EC0142

PART A**[70 MARKS]****QUESTION 1****[4 Marks]**

Program 1 is the program that is used by the parking attendant to calculate the fee for parking based on the hours and rates. The parking building consists of employee and non-employee car with different type of rates:

- (a) Employee of the company: RM 1.50 per hour
- (b) Non-employee: RM 3.00 per hour

Based on the comments given in **Program 1**, fill in the blank lines with appropriate C++ statements. The program should generate the output as shown in **Figure 1**. *Note:*

The **bold**

texts in **Figure 1** indicate input from the user.

(4 marks)

```
The number of cars: 2
Hours parked for car 1: 3
Hourly pay rate for car 1: 1.50

Hours parked for car : 5
Hourly pay rate for car : 3.00

Here is the total fee for each car:
Car [1] = RM4.50
Car [] = RM15.00
```

Figure 1: Example runs of the Program 1

```
1  // Program 1
   #include<iostream>
3  #include<iomanip>
4  using namespace std;
5
6  int main()
7  {
8      const int no_cars = ; //Constant for the array size
9
10     //Declare an array named hours and rate to allocate 2
11     //elements of type integer and double respectively.
12     //Use constant declared in line 8.
```

```

13 int hours[no_cars];
14 double rate[no_cars];
15
16 cout << "The number of cars: " << no_cars;
17
18 cout << endl;
19
20 for (int index = 0; index < no_cars; index++)
21 {
22     cout << "\nHours parked for car " << (index+1) << ": ";
23     //Read a value and assign it into array named hours
24     cin >> hours[index];
25     cout << "Hourly pay rate for car " << (index+1) << ": ";
26     //Read a value and assign it into array named rate
27     cin >> rate[index];
28 }
29
30 cout << "\nHere is the total fee for each car:\n";
31 cout << fixed << showpoint << setprecision();
32
33 for (int index = 0; index < no_cars; index++)
34 {
35     //Calculate a value of totalPay (fee for parking)
36
37     double totalPay = hours[index] * rate[index];
38     cout << "Car [" << (index + 1);
39     cout << "] = RM" << totalPay << endl;
40 }
41
42 return 0;
43 }

```

QUESTION 2

[16 Marks]

Given the declaration and initialization of some parallel arrays as in the following code segment:

```
string patients[ ] = {"Wendy", "Kumar", "Ros", "Mael"};  
int age[ ] = {5, 43, 3, 54};  
double averageSugarLevel[4];
```

Based on the concept of arrays and using loops, answer the following questions:

- (a) Define a two-dimensional array named **sugarLevel** which has large enough space to hold the data in the following table. Initialize the array with values from the table.

(2 marks)

4.5	5.7	6.8
6.2	7.1	7.3
5.1	6.3	8.0
7.5	8.5	7.3

Answer:

```
double sugarLevel
```

```
[4][3]={ {4.5,5.7,6.8}, {6.2,7.1,7.3}, {5.1,6.3,8.0}, {7.5,8.5,7.3} }; // an array with 4  
rows & 3 columns
```

- (b) Write a **code segment** that prints the patient names along with their ages. The output should be as follows:

(3 marks)

Age of Patients	

Wendy	5
Kumar	43
Ros	3
Mael	54

Answer:

```
int main ( )
{
    const int SIZE=4;
    string patients [SIZE]={"Wendy","Kumar","Ros","Mael"};
    int age[SIZE]={25,43,32,54};

    cout<<"Age of Patients\n";
    cout<<"-----\n";
    for(int i=0;i<SIZE;i++)//assign element by element
    {
        cout<<patients[i]<<"\t";//print patient name
        cout<<age[i]<<"\n";//print patient age
    }
    return 0;
}
```

(c) Write a **code segment** that produces the following output from the array named **sugarLevel** that you defined in (a).

Name	Breakfast	Lunch	Dinner
Wendy	4.5	5.7	6.8
Kumar	6.2	7.1	7.3
Rose	5.1	6.3	8.0
Mael	7.5	8.5	7.3

Answer:

```
int main( ) {
    const int SIZE=4;
    double sugarLevel [4][3]={ {4.5,5.7,6.8},{6.2,7.1,7.3},{5.1,6.3,8.0},{7.5,8.5,7.3}}; //
    an array with 4 rows & 3 columns
    string patients [SIZE]={"Wendy","Kumar","Ros","Mael"};
    double(*s)[3];
```

```

cout<<"Name\t"<<"Breakfast\t"<<"Lunch\t"<<"Dinner\n";
for(int i=0;i<4;i++){ //print patient names and sugarLevel accordingly
{
    s=&sugarLevel[i];
    cout<<patients[i]<<"\t";
}
for (int j=0;j<1;j++){
cout<<*(*s+j)<<"\t\t"<<*(*s+1)<<"\t"<<*(*s+2)<< fixed << setprecision(1)<<endl;
} //pointer to 2 dimensional array //set pointer to 1 decimal places
}
return 0;
}

```

(d) Write a **code segment** to calculate the average sugar level of the patient. The code also store the average sugar level value in the array **averageSugarLevel1** and print the list of the patients' average sugar level value as follows. (7 marks)

Sugar Level of Patients	

Wendy	25
Kumar	43
Ros	32
Mael	54

Answer:

```
int main (){
const int SIZE=4;
double sugarLevel [4][3]={ {4.5,5.7,6.8}, {6.2,7.1,7.3}, {5.1,6.3,8.0}, {7.5,8.5,7.3} }; // an array
with 4 rows & 3 columns
string patients [SIZE]={"Wendy","Kumar","Ros","Mael"};
int age[SIZE]={25,43,32,54};
double averageSugarLevel [4];

cout<<"Sugar Level of Patients\n";
cout<<"-----\n";
for(int i=0;i<4;i++){
    {
        cout<<patients[i]<<"\t";//print patient name
    }
    for (int j=0;j<3;j++){
        averageSugarLevel[i]+=sugarLevel[i][j]/3;//calculate average sugar per patient
        if(j==2)//print average sugar level in column 2
            cout<<averageSugarLevel[i]<< fixed<<showpoint << setprecision(1)<<endl;
    }
}
return 0;
}
```

QUESTION 3

[11 Marks]

- (a) Name at least two member functions associated with **iostream** object. (2 marks)

Answer:

- cin and cout

- (b) Suppose two file stream objects named **fin** and **fout** have been declared in the code as follows:

```
ifstream
fin;
ofstream
fout;
```

Complete the code so that the program can copy 10 numbers from a file named **file1.dat** and save the numbers to another file named **file2.dat** (copy 10 numbers between files). The program should also include an error checking for opening the files. (9 marks)

Answer:

```
#include<iostream>
#include<sstream>
#include<fstream>
#include<iomanip>

using namespace std;

void createfile();

int main ()
{
    createfile();

    string line; //For writing text file

    ifstream fin ("file1.dat");
    ofstream fout ("file2.dat");

    if(fin && fout){

        while(getline(fin,line)){
            fout << line << "\n"; //copies file1 to file2
        }

        cout << "Copying...\n";
```

```

        cout << "Copy Finished \n";
    } else {
        printf("Cannot read File"); //Something went wrong
    }

    //Closing file
    fin.close();
    fout.close();

    return 0;
}
void createfile(){
    int num;
    string line;
    ifstream fin ("file1.dat");
    ofstream fout;

    fout.open("file1.dat");
    cout<<"Now writing information to a file. \n"; //Write four great names to the file.

    for (int i= 1; i<= 10; i++){
        fout<<" "<<i;
    }

    fout.close(); //close the file
    cout<<"Done.\n";
}

```

QUESTION 4**[4 Marks]**

Based on the **Program 2**, write the output into **Table 1**. *Note*: one cell represents one space per digit or character. (4 marks)

```
1 // Program 2
2 //This program write three rows of numbers to a file
3 #include <iostream>
4 #include <fstream>
5 #include <iomanip>
6 using namespace std;
7
8 int main ()
9 {
10     const int ROWS = 3;
11     const int COLS = 3;
12     double nums [ROWS][COLS] = {1501, 3.579, 984,
13                                   5.3, 91, 1979,
14                                   4, 606, 11.087};
15     fstream outFile ("Jadual.txt", ios::out);
16
17     for (int row = 0; row < ROWS; row++)
18     {
19         for (int col = 0; col < COLS; col++)
```

```

20      {
21          outFile << setprecision (1) << fixed << showpoint;
22          outFile << setw(6) << nums [row][col];
23      }
24      outFile << endl;
25  }
26  outFile.close();
27  return 0;
28  }

```

Table 1: Output file named **Jadual.txt**

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

QUESTION 5**[4 Marks]**

As you are travelling to the United States, it is important to know the temperature for you to be able to prepare what to wear for the day. Apparently in the United States, they use Fahrenheit and not Celsius. A conversion program that will ask for a temperature in Fahrenheit, convert that value to Celsius, and display the result, would definitely become handy. Complete the **Program 3**. You need to write the function **convert**. *Note: $C = 5 / 9 (F - 32)$* . Do not modify the function main.

1	// Program 3
2	#include <iostream>
3	#include <iomanip>
4	using namespace std;
5	
6	//Write your function prototype here
7	void convert (double&);
8	int main ()
9	{
10	double temperature;
11	
12	cout << "Enter a temperature in Fahrenheit, and I will"
13	<< "convert it to Celsius: ";
14	cin >> temperature;
15	convert (temperature);
16	cout << fixed << setprecision(4);
17	cout << "Value in Celsius: " << temperature << endl;
18	
19	return 0;
20	}
21	
	//Write the function convert here
3	void convert (double &temp)
4	{
5	temp = 5.0 / 9.0 * (temp - 32.0) ;
6	return;
7	}

QUESTION 6[10 Marks]

(a) Determine the output and draw a memory layout (or memory allocation) of the pointers and variables for code segment below: (7 marks)

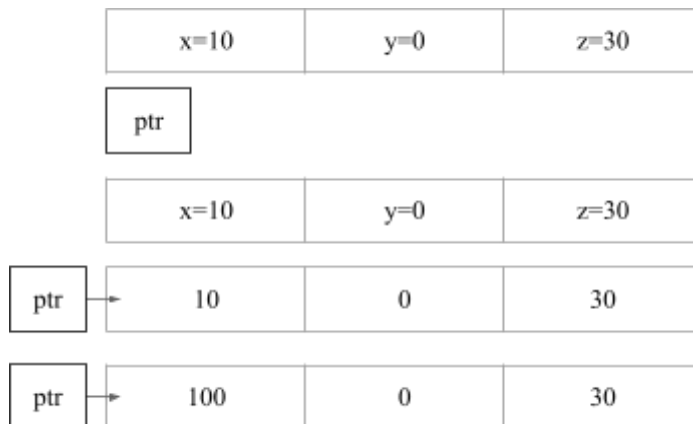
Note: Draw a memory layout that represents C++ statement line by line.

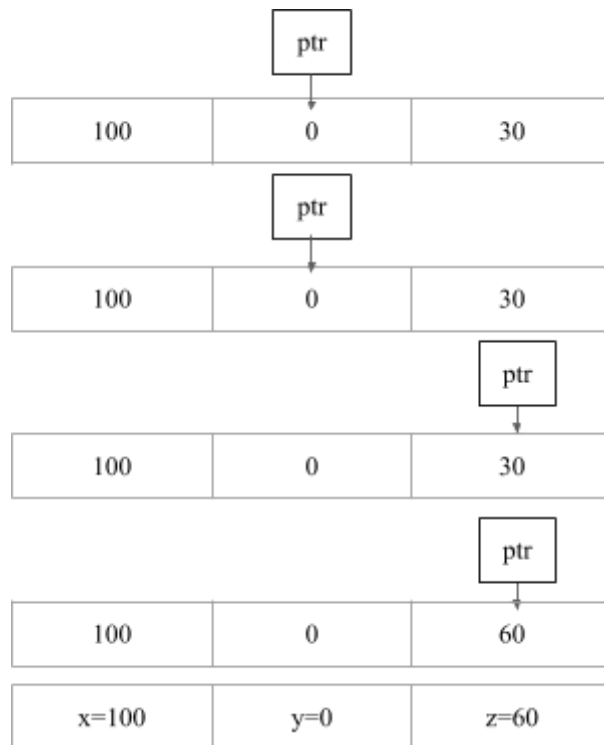
```
int    x = 10, y = 0, z = 30;
int *ptr;

cout << x << " " << y << " " << z << endl;
ptr = &x;
*ptr *= 10;
ptr = &y;
*ptr *= 4;
ptr = &z;
*ptr *= 2 ;

cout << x << " " << y << " " << z << endl;
```

Answer:





Output:

10 0 30
100 0 60

(b) Determine the output for code segment below:

(3 marks)

```
int numbers[5] = {2,4,6,8,10};
int *numPtr;

numPtr = numbers;
cout << *(numbers + 3) << endl;
cout << *(numbers) << endl;

for (int count=0; count < 2; count++)
{
    cout << *numPtr << "\\t";
    numPtr++;
}
cout << endl;

for (int count=0; count < 2 ; count++)
{
    numPtr--;
    cout << *numPtr << "\\t";
}
```

Answer:

Output:

```
8
2
2  4
4  2
```

QUESTION 7

[5 Marks]

The code segment below has five (5) syntax and/ or logical errors. Identify and describe each of the errors. (5 marks)

```
struct TwoLoc
{
    int x = 10;
    int y = 15;
}

int main()
{
    TwoLoc poi nt[10];
    point.x[0] = 1;
    point.y[0] = 1;
    cout << point << endl;
    return 0;
}
```

Answer:

```
struct TwoLoc
{
    int x=10;
    int y=15;
}; //must have ; after closing }
```

```
int main()
{
```

```
TwoLoc point[1]; //to save the memory better to put lower array size
point[0].x=1; //array size
point[0].y=1; //array size
cout<<point[0].x<<" "<<point[0].y<<endl; //use dot operator to refer to member of
structure
return 0;
}
```

QUESTION 8**[16 Marks]**

Meteorological Department is responsible to keep the information about weather forecast in Malaysia and display the information through the website. Write the code segment for each of the following tasks:

- (a) i) Declare a structure type named **TempScale**, with the following members: **fahrenheit** : a double value
celsius : a double value
- ii) Declare a structure type named **Reading**, with the following members: **windSpeed** : an integer value
humidity : a double value
temperature: a **TempScale** structure variable
- iii) Declare a variable of structure type **Reading** named **today**. (6 marks)

Answer:

- (a) i) struct TempScale
{
 double fahrenheit, celcius; // Temp. in fahrenheit and celcius
};
- ii) struct Reading
{
 int windSpeed; // Wind speed
 double humidity; // Humidity
 TempScale temperature; // Nested structure
};
- iii) Reading today;

- (b) Write statements that will store the following data into the variable you declared in (a).

<i>Wind speed: 8 km/h</i> <i>Humidity: 53%</i> <i>Fahrenheit temperature: 80.2 degrees</i> <i>Celsius temperature: 26.8 degrees</i>
--

(3 marks)

Answer:

```
b) today.windSpeed = 8;  
   today.humidity = 53;  
   today.temperature.fahrenheit = 80.2;  
   today.temperature.celcius = 26.8;
```

- (c) Define a function named **showReading**. It should accept a **Reading** structure variable as its argument. The function should display the contents of the variable onto the screen.

- (c) void showReading(Reading t)

```
{  
    cout << "Malaysia today's weather. " << endl << endl;  
    cout << "Wind Speed: " << t.windSpeed << " km/h" << endl;  
    cout << "Humidity: " << t.humidity << "%" << endl;  
    cout << "Fahrenheit temperature: " << t.temperature.fahrenheit << " degrees" <<  
    endl;  
    cout << "Celcius temperature: " << t.temperature.celcius << " degrees" << endl;  
}
```

Full Program :

```
//This is a program that responsible to keep information about weather forecast in
//Malaysia.
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;

// (a) i) The TempScale holds data about today's temperature at Malaysia.
struct TempScale
{
    double fahrenheit, celcius; // Temp. in fahrenheit and Celsius
};

// (a) ii) Structure type declaration named Reading.
struct Reading
{
    int windSpeed; // Wind speed
    double humidity; // Humidity
    TempScale temperature; // Nested structure
};

void showReading(Reading); // Argument passed by value

int main()
{
    Reading today; // (a) iii) ) Declaration of variable of structure type Reading named
    //today.

    // (b) To store data into the variable that had been declared in (a).
    today.windSpeed = 8;
    today.humidity = 53;
    today.temperature.fahrenheit = 80.2;
```

```

    today.temperature.celcius = 26.8;

    showReading(today);
}

// (c) Definition of function showReading
void showReading(Reading t)
{
    cout << "Malaysia today's weather. " << endl << endl;
    cout << "Wind Speed: " << t.windSpeed << " km/h" << endl;
    cout << "Humidity: " << t.humidity << "%" << endl;
    cout << "Fahrenheit temperature: " << t.temperature.fahrenheit << " degrees" <<
    endl;
    cout << "Celcius temperature: " << t.temperature.celcius << " degrees" << endl;
}

```

Answer for Part B

```
#include<iostream>
#include<fstream>
#include<iomanip>
using namespace std;

//Global constants
const int NUM_STATE = 14;
const int NUM_YEAR = 10;
//global variable
ofstream out("output.txt");

//struct defined in part (a)
struct dataAcc{
    int numAcc[10];
    string state;
    float avg;
};

//displayLine function
void displayLine(){
    for(int i=0;i<98;i++){
        cout<<"-";
        cout<<endl;
    }
    float cal_Avg(int arr[10]){
        float sum=0,avg;
        for(int i=0;i<10;i++){
            sum+=arr[i];    //calculate sum of 10 elements in array
        }
        avg = sum/10;        //average = sum/10
        return avg;        //return average
    }

int find_HighLow(struct dataAcc data[50],int size){
    //variables for highest and lowest accidents, their years and states
    int high=0,low=999999;
    int high_year,low_year;
    string high_state,low_state;
    //calculate highest and lowest accidents
    for(int i=0;i<size;i++){
        for(int j=0;j<10;j++){
            //if it is lower than the current lowest
            if(data[i].numAcc[j]<low){
                //update the data
                low = data[i].numAcc[j];
                low_state = data[i].state;
                low_year = i+2006;
            }
            //if it is higher than the current highest
            if(data[i].numAcc[j]>high){
```

```

        //update the data
        high = data[i].numAcc[j];
        high_state = data[i].state;
        high_year = i+2006;
    }
}
}
//print the lines and the highest and lowest data
displayLine();
cout<<"The highest number of road accidents = "<<high<<" at "<<high_state<<" on
"<<high_year<<endl;
displayLine();
cout<<"The lowest number of road accidents = "<<low<<" at "<<low_state<<" on
"<<low_year<<endl;
displayLine();
}
int main(){

    dataAcc *ptr;
    ifstream f("input.txt");

    ptr = new dataAcc [NUM_STATE];

    for (int i = 0; i < NUM_STATE; i++)
    {
        for(int j = 0; j<10;j++){
            f>> ptr[i].numAcc[j];
        }
        getline(f, ptr[i].state);
    }

    ofstream out("output.txt");
    out<<" STATE
\t\t2006\t\t2007\t\t2008\t\t2009\t\t2010\t\t2011\t\t2012\t\t2013\t\t2014\t\t2015\t\tAVERAGE\n
";
    for (int i = 0; i < NUM_STATE; i++){
        out <<ptr[i].state<<" \t\t";

        for(int j = 0; j<10;j++){
            out<<ptr[i].numAcc[j]<<"\t\t";
        }
        out<<cal_Avg(ptr[i].numAcc)<<endl;
    }
    cout<<"Done creating output.txt\n";
    displayLine();
    cout<<" STATE \t 2006\t 2007\t 2008\t 2009\t 2010\t 2011\t 2012\t 2013\t 2014\t
2015\t AVERAGE\n";
    displayLine();
    for (int i = 0; i < NUM_STATE; i++){
        cout <<ptr[i].state<<" \t\t";

```

```

        for(int j = 0; j<10;j++){
            cout<<ptr[i].numAcc[j]<<"\t";
        }
        cout<<" "<<cal_Avg(ptr[i].numAcc)<<endl;
    }
    struct dataAcc data[50];
    out<<find_HighLow(data,NUM_STATE);

    return 0;
}

```