



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF SOCIAL SCIENCES AND
HUMANITIES

SECJ1013 - 05 (PROGRAMMING TECHNIQUE I)

SEMESTER 1, 2020/2021

ASSIGNMENT 4

LECTURER NAME:

Dr. GOH EG SU

GROUP MEMBER:

NAME	MATRIC NO.
Lee Jia Xian	A20EC0200
Shingthai Srisoi	A20EC0147
Tan Yong Sheng	A20EC0157

Part A

Question 1

```
1  // Program 1
2  #include<iostream>
3  #include<iomanip>
4  using namespace std;
5
6  int main()
7  {
8      const int no_cars = 2; //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
14     int hours[no_cars];
15     double rate[no_cars];
16
17     cout << "The number of cars: " << no_cars;
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 >> cin >> hours[index];
25
26         cout << "Hourly pay rate for car " << (index+1) << ": ";
27         //Read a value and assign it into array named rate
28         cin >> cin >> rate[index];
29     }
30     cout << "\nHere is the total fee for each car:\n";
```

```

cout << fixed << showpoint << setprecision(2);

for (int index = 0; index < no_cars; index++)
{
    //Calculate a value of totalPay (fee for parking)

    double totalPay = hours[index] * rate[index];

    cout << "Car [" << (index + 1);
    cout << "] = RM" << totalPay << endl;
}
} return 0;

```

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[] = {25, 43, 32, 54};

double averageSugarLevel[4];

```

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

- (a) Define a two-dimensional array named **sugarLevel1** 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} };
```

- (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      25
Kumar      43
Ros        32
Mael       54
```

Answer :

```
cout << "Age of Patients"<<endl;
cout <<"-----"<<endl;
cout<<left;

for(int row=0;row<4;row++){
    cout <<setw(10)<<patients[row]<<age[row]<<endl;
};
```

- (c) Write a **code segment** that produces the following output from the array named **sugarLevel** that you defined in (a). (4 marks)

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:

```
cout<<left<<fixed<<setprecision(1);
cout <<setw(15) <<"Name" <<setw(12)
<<"Breakfast"<<setw(10)<<"Lunch"<<"Dinner"<<endl;

for(int row=0;row<4;row++){
    cout << setw(18) <<patients[row]<<setw(12);
    for(int col=0;col<3;col++){
        cout <<setw(10)<<sugarLevel[row][col];
    }
    cout <<endl;
}
```

- (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 **averageSugarLevel** 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:

```
for(int row=0;row<4;row++){  
    for(int col=0;col<3;col++){  
        averageSugarLevel[row]+= sugarLevel[row][col];  
    }  
    averageSugarLevel[row]/=3;  
}  
  
cout <<"Sugar Level of Patients"<<endl;  
cout <<"-----\n";  
  
for(int row=0;row<4;row++){  
    cout <<left<<fixed<<setprecision(1);  
    cout <<setw(10) <<patients[row] <<averageSugarLevel[row] <<endl;  
}
```

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 **f_{in}** and **f_{out}** 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<iomanip>
```

```
#include<fstream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    ifstream fin;
```

```
    ofstream fout;
```

```
    fin.open("file1.txt"); //open file1
```

```
    fout.open("file2.txt");//open file2
```

```
    int num;
```

```
//checking for opening file  
if(!fin) cout << "ERROR! cannot open file1.dat\n";  
if (!fout) cout <<"ERROR! cannot open file2.dat\n";
```

```
//copy num between files  
for(int n=0;n<10;n++){  
    fin >> num;  
    fout << num<<endl;  
}
```

```
//close the files  
fin.close();  
fout.close();
```

```
return 0;  
}
```


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                                   24, 606, 11.0287};
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	4	.	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: $C = 5/9 \times (F - 32)$* . Do not modify the function **main**. (4 marks)

```
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
9  int main ()
10 {
11     double temperature;
12
13     cout << "Enter a temperature in Fahrenheit, and I will
14         << convert it to Celsius: ";
15     cin >> temperature;
16     convert (&temperature);
17     cout << fixed << setprecision(4);
18     cout << "Value in Celsius: " << temperature << endl;
19
20     return 0;
21 }
22
23 //Write the function convert here
24 void convert (double &f){
25     f=(5.0/9)*(f-32);
26 }
```

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 = 20, 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 :

```
int x = 10, y = 20, z = 30; int *ptr;
```

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

```
ptr = &x;
```

x 10 y 20 z 30



```
*ptr=10
```

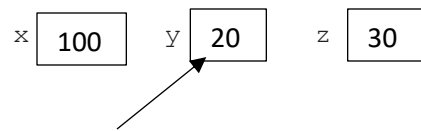
```
*ptr *= 10;
```

x 100 y 20 z 30



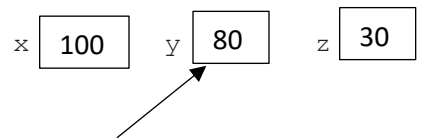
```
*ptr=10*10=100
```

```
ptr = &y;
```



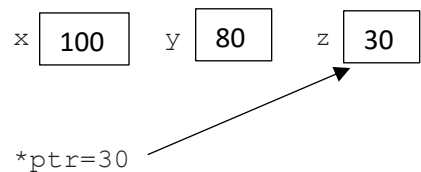
```
*ptr=20
```

```
*ptr *= 4;
```



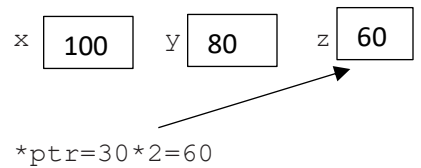
```
*ptr=20*4=80
```

```
ptr = &z;
```



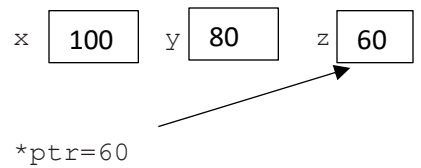
```
*ptr=30
```

```
*ptr *= 2;
```



```
*ptr=30*2=60
```

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



```
*ptr=60
```

Output:

10 20 30

100 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:

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 point[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 struct declaration function
```

```
int main()
{
```

Better to put lower array size to save the memory

```
    TwoLoc point[1];

    //wrong declaration

    point[0].x= 1;

    point[0].y= 1;

    cout << point[0].x <<point[0].y <<endl; //must put subscript
    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:

i)
struct TempScale
{
double fahrenheit, celsius;
};

ii)
struct Reading{
int windSpeed;
double humidity;
TempScale temperature;
};

iii) Reading today;

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

Wind speed: 8 km/h

Humidity: 53%

Fahrenheit temperature: 80.2 degrees

Celsius temperature: 26.8 degrees

Answer :

```
today.windSpeed = 8;  
today.humidity = 53;  
today.temperature.fahrenheit=80.2;  
today.temperature.celsius = 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. (7 marks)

Answer:

```
void showReading (Reading today){  
    cout << "Wind speed: " <<today.windSpeed<<" km/h\n";  
    cout << "Humidity: " <<today.humidity <<" %\n";  
    cout << "Fahrenheit temperature: "<<today.temperature.fahrenheit<<" degrees\n";  
    cout << "Celsius temperature: "<<today.temperature.celsius<<" degrees\n";  
}
```

Part B

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

//global constant 10 years and 14 states in Malaysia
const int NUM_STATE=14,NUM_YEAR=10;
//global ofstream
ofstream out;

//structure for accident data
struct dataAcc{
    int numAcc[10];//total accident in 10 year
    string state;//name of a state
    float avg;//average accident 2006-2015
};

//display 98 '-' to draw a line in output.txt
void displayLine(){
    for(int i=0;i<98;i++)
        out<<"-";
    out<<endl;
}

//calculate average
double cal_Avg(int acc[NUM_YEAR]){
    int sum=0;
    for(int i=0;i<NUM_YEAR;i++){
        sum+=acc[i];
    }
    double avg = static_cast<double>(sum)/NUM_YEAR;
    return avg;
```

```
}
```

```
//find the highest cases in 14 states between 10 years
```

```
void find_HighLow(dataAcc data[NUM_STATE]){
```

```
    int highest=0,year;
```

```
    string highState;
```

```
    for(int i=0;i<NUM_STATE;i++){
```

```
        for(int j=0;j<NUM_YEAR;j++){
```

```
            if(data[i].numAcc[j] > highest){
```

```
                highest=data[i].numAcc[j];
```

```
                highState=data[i].state;
```

```
                year=2006+j;
```

```
            }
```

```
        }
```

```
    }
```

```
    //print in output file
```

```
    out<<"The highest number of road accident = "<<highest<<" at "<<highState;
```

```
    out<<" on "<<year<<endl;
```

```
}
```

```
int main(){
```

```
    //declare dataAcc array for 14 states
```

```
    dataAcc data[NUM_STATE];
```

```
    //declare infile and open file for input and output
```

```
    ifstream load;
```

```
    load.open("input.txt");
```

```
    out.open("output.txt");
```

```
    //check if succesfully open the file
```

```
    if(load.is_open() && out.is_open()){
```

```
        cout<<"Successfully read the file\n";
```

```
    }else{
```

```
        cout<<"ERROR: cannot open file\n";
```

```
}
```

```
//read data
```

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

```
for(int i=0;i<NUM_STATE;i++){
```

```
    for(int j=0;j<10;j++){
```

```
        load>>data[i].numAcc[j];
```

```
    }
```

```
    cin.ignore(0);
```

```
    getline(load,data[i].state);
```

```
}
```

```
//print data
```

```
displayLine();
```

```
out<<left<<setw(20)<<"    STATE";//print state
```

```
for(int i=2006;i<2016;i++){
```

```
    out<<setw(7)<<i;//print year
```

```
}
```

```
out<<"AVERAGE\n";
```

```
displayLine();
```

```
//outer loop for looping each states
```

```
for(int i=0;i<NUM_STATE;i++){
```

```
    //print states
```

```
    out<<setw(17)<<left<<data[i].state;
```

```
    for(int j=0;j<NUM_YEAR;j++){
```

```
        //print number of accident of the state
```

```
        out<<right<<setw(7)<<data[i].numAcc[j];
```

```
    }
```

```
    //print average accident for the state
```

```
    out<<setw(10)<<right<<fixed<<setprecision(1)<<cal_Avg(data[i].numAcc)<<endl;
```

```
}
```

```
displayLine();  
find_HighLow(data);//print highest  
displayLine();  
cout<<"DONE!!!";  
  
//close file  
load.close();  
out.close();  
  
return 0;  
}
```

