UTM

UNIVERSITI TEKNOLOGI MALAYSIA

UNIVERSITI TEKNOLOGI MALAYSIA (UTM)

PROGRAMMING TECHNIQUE – 1 (03)

ASSIGNMENT 2

Prepared by:

NAME	MATRIC ID
HASSAN MUSTAFA IBRAHIM	A20EC4025
MUHAMMAD IMRAN HAKIMI BIN MOHD SHUKRI	A20EC0213
MOHAMED TAREK ELSAYED MOHAMED YOUSEF MOHAMED TORKY	A20EC0275

Submitted to: Dr. Goh Eg Su

Submitted on: 23rd December 2020

SECTION A – TRUE/FALSE QUESTIONS [Total 8 marks]

There are **FOUR (4)** questions in this section. For each statement given in this section, identify whether the statement is **TRUE/FALSE** and write your answer with your **reason** in the space given. Each question carries **2** marks.

1. The following C++ code in Figure 1 determines the amount of interest based on user's amount value in his/her account. If the amount value in his/her account falls under option 2, the program will only calculate interest = 2.6 for that particular's user account.

```
switch (account_value)
{
      case 1:
          interest = 2.3;
          break;
      case 2:
          interest = 2.6;
      case 3:
          interest = 2.9;
          break;
      default:
          interest = 0.0;
}
```

Figure 1

Answer : False

Reason : When the amount value is 2, the interest will be set to 2.6, and since there is no break statement it will continue to execute the next case and will set the interest to 2.9 and will stop because there's break statement.

2. If **x=3** and **y=2**, the following segment program in Figure 2 will produce 0 when it is executed.

```
int main()
{
      int x, y;
      cout<<" Please enter two numbers: ";
      cin>> x>>y;

      for (int i=1;i<=y;i++)

          for (int j=1;j<=x;j +=2)
          {
               cout<<setw(2)<< y/x;
          }
      system ("PAUSE");
      return 0;
}
```

Figure 2

Answer : True

Reason : ~~Because y and x are declared as int and not double. When the output is in decimal, the compiler will only take the front number before the decimal. In this case, the result for y/x is 0.666 and the output taken is 0 only. (integer division)~~

3. The `terminate()` function causes a program to terminate, regardless of which function or control mechanism is executing.

Answer : False

Reason : The exit() function causes the program to terminate regardless of which function or control mechanism is executing, not terminate() function.

4. The output of the following program in Figure 3 is as illustrated in Figure 4.

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

int call(int x){
  static int a = 10;
  int b;
   b=x;
   a+=b++;
   return a;
  }

int main()
{

  cout << "The value return is " << call(5) << endl;
  cout << "The value return is " << call(12) << endl;
  system("pause");
  return 0;

}
```

Figure 3

```
The value return is 15
The value return is 27
Press any key to continue . . .
```

Figure 4

Answer : True

Reason : The variable a is static variable, so it will retain it's last value, which is 15

SECTION B – STRUCTURE QUESTIONS [Total marks 50]

There are **FOUR (4)** structured questions. Answer all questions in the space provided. The marks for each part of the question is as indicated.

1. (a) Based on the information displayed in Figure 5, complete the missing C++ (Figure 6) code by writing a ternary conditional operator. [5 marks]

```
Enter 'y' or non-'y':
y
If 'y', answer is = 1
Press any key to continue . . .
```

```
Enter 'y' or non-'y':
p
Else, answer is = 0
Press any key to continue . . .
```

Figure 5

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

int main()
{
      // initialize x to a dummy value
      char x = 'n';
      cout << "Enter 'y' or non-'y': " << endl;
      cin >> x;
      // write the output using the ?: operator

      int answer= x=='y'?  1 :  0;

      cout << answer;
      cout << endl;
      system("pause");
      return 0;
}
```

Figure 6

(b) Consider the following output screen in Figure 7 that shows input and output of a banking activity. When deposit is chosen, the latest balance is the addition of old balances with the amount deposited, and when withdrawal is chosen, the latest balance is the subtraction of the old balance with the amount deposited. When any other character is entered, you should let the user know that the code is not allowed, and he/she must try again. Assuming your current balance in the account is RM300, write a complete C++ code for the following program by using 'switch' case statements. [8 marks]

```
Enter your transaction code, d - deposit, w - withdrawal:
d
Enter amount RM150
Your current balance is now RM 450
```

Figure 7

Answer 1(b):

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

int main() {
    char selection;
    double amount, balance=300;
    cout <<"Enter your transaction code, d - deposit, w - withdrawal: ";
    cin >> selection;


    switch (selection)
    {
       case 'd':
            cout << "Enter transaction amount RM : ";
            cin >> amount;
            balance += amount;
            break;
       case 'w':
            cout << "Enter transaction amount RM : ";
            cin >> amount;
            balance -= amount;
            break;
       default :
            cout << "The service asked is not available. Please try again.\n\n";
            return main();
    }
    cout << "Your current balance is now RM "<<balance<<endl;
    return 0;
}
```

Answer 1(b):

2.

```
int main()
{
int a, x =0;
cout<<" please enter  a number ";
cin>> a;

if (a == 1 || a==2)
   x ++;
else if ( a==3 || a==4)
     x--;
else cout << x+=2;
     cout << x;
system ("PAUSE");
return 0;
}
```

Figure 8

(a) Based on C++ code in Figure 8, do the following amendment as follows:

(i) Convert the if statement to a switch-case statement.

(ii) Also, write an input validation loop that asks the user to enter a number in the range of 1 through 4. [7 marks]

Answer 2(a):

(i)

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

int main()
{
int a, x =0;
cout<<" please enter  a number ";
cin>> a;

switch (a)
{
   case 1 :
   case 2 :
      x++;
      break;
   case 3 :
   case 4 :
      x--;
      break;
   default  :
      x+=2;
      break;
}
cout << x;
return 0;
}
```

Answer 2(a):

(ii)

```
int a, x =0;
do{
cout<<"Please enter  a number only between 1 and 4.\n ";
cin>> a;
}
while (a<1 ll a>4);
```

3. You are required to develop a program to compute and display the charges for patients of Hospital Tun Aminah Johor.

(a) Write a function to get the inputs from user. The function should accept reference arguments in order to access the parameters in the **`main()`** function. The function should ask for the following attributes in Table 1: [4 marks]

Table 1

Patient Attributes
Number of days spent
Daily room rate
Medication charges
Service charges

Answer 3(a):

```
void information(double &numDaysSpent,
                 double &dailyRoomRate,
                 double &medicationCharges,
                 double &serviceCharges)
```

Answer 3(a):

```
{
    cout << "How many days spent? ";
    cin >> numDaysSpent;

    cout << "How much the room daily rate? ";
    cin >> dailyRoomRate;

    cout << "How much the medication charges? ";
    cin >> medicationCharges;

    cout << "How much for the service charges? ";
    cin >> serviceCharges;
}
```

(b) There are two types of patient in Hospital Tun Aminah Johor as described in Table 2. In the program, patient's type is declared as global variable. The following formula in Table 2 is used to compute the total charges. Write two overloaded function to calculate the total charges. One of the functions should accept arguments for the in-patient, while the other function accepts arguments for out-patient. Both functions should return the total charges. [5 marks]

Table 2

Type	Formula
In-patient	Total charges = Number of days spent * Daily room rate + Medication charges + Service charges
Out-patient	Total charges = Medication charges + Service charges

Answer 3(b):

```
double totalCharges(double numDaysSpent,
                    double dailyRoomRate,
                    double medicationCharges,
                    double serviceCharges)
{
  double total = numDaysSpent * dailyRoomRate + medicationCharges + serviceCharges;
  return total;
}

double totalCharges(double medicationCharges, double serviceCharges)
{
  double total = medicationCharges + serviceCharges;
  return total;
}
```

Answer 3(b):

4. (a) Based on the Program in Figure 9, what is the output displayed when the program is executed? [2 marks]

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

int two(int n)
{
    int ans;
    if (n==1)
       ans=0;
    else
        ans = 1+two(n/2);
    return ans;
}

int main()
{
    int y;

    y = two(13);
    cout<< y;
    system("pause");
    return 0;
}
```

Figure 9

Answer 4(a):

The output is

3

(b) What is the output of the above program (Figure 10) if the user enters **12** and **14**?

[9 marks]

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

void func1(int = 5, int = 6);
void func2(int &, int &, int &);
void func3(int&, int&);
void func4(int,int,int);

int main()
{
    int x=0, y=0,z=0;
    func1();
    cout<<x <<" "<<y<<" "<<z<<endl;
    func3(x,y);
    cout<<x <<" "<<y<<" "<<z<<endl;
    func2(x,y,z);
    cout<<x <<" "<<y<<" "<<z<<endl;
    func4(x,y,z);
    cout<<x <<" "<<y<<" "<<z<<endl;
    system("pause");
    return 0;
}

void func1(int a, int b)
{
     a++;
     b+=a;
     cout<<a<<" "<<b<<endl;
}

void func2(int &a, int&b, int&c)
{
  b++;
     c--;
     a=b+c;
}

void func3(int &a, int&b)
{
     cout<<"Enter two numbers: ";
     cin >> a >> b;
}

void func4(int a, int b, int c)
{
     a=b-c;
     cout <<a << " " << b <<" " << c << endl;
}
```

Figure 10

Answer 4(b):

6 12

0 0 0

Enter two numbers:12 14

12 14 0

14 15 -1

16 15 -1

14 15 -1

(c) Given the following excerpted program (Figure 11):

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

int input();
int totalDay(int);
double calcAvg(int, int);

int main()
{
    int x;
    double avg;
    x = input();
    avg = calcAvg(x,totalDay(x));
    cout<<fixed<<showpoint<<setprecision(2);
    cout<<avg;
    system("pause");
    return 0;
}
```

Figure 11

(i) Based on the **`main()`** function in Figure 11, you are required to write the following user-defined functions: [10 marks]

Table 3. User-defined functions

Function	Description
`input`	This function asks the user for the number of employees in the company. This value should be returned as an **`int`**. The function accepts no argument.
`totalDay`	This function accepts one argument: the number of employees in the company. The function should ask the user to enter the number of days each employee absent during the past year. The total of these days should be returned as an **`int.`**
`calcAvg`	This function accepts two arguments: the number of employees in the company and the total number of days absent for all employees during the year. The function should return, as a **`double`**, the average number of days absent. This function does not display any outputs and does not ask the user for input

Answer 4(c-i):

```
int input (){
    int num_emp;
    cout << "Enter the number of employees in the company: ";
    cin >> num_emp;
    return num_emp;
}
```

Answer 4(c-i):

```
int totalDay (int x){

    int days, sumdays=0;

    for(int i = 0; i < x; i++)
    {
        cout << "Enter the number of days " << i+1 <<" employee was
absent during the past year : ";
        cin>>days;
        sumdays+=days;
    }
    return sumdays;
}

double calcAvg (int num_emp, int totalDay){
   return (static_cast<double>(totalDay)/num_emp);
}
```

Answer 4(c-i):