CONFIDENTIAL

UNIVERSITI TEKNOLOGI MALAYSIA

FINAL EXAMINATION

SEMESTER I 2011/2012

SUBJECT CODE	**: SCJ1013 / SCJS1013**
SUBJECT NAME	**: PROGRAMMING TECHNIQUE I**
YEAR/COURSE	**: 1 (SCJ / SCV / SCB / SCD/ SCI / SCR)**
TIME	**: 2 ½ HOURS**
DATE	**: 13 JANUARY 2012**
VENUE	**: N28 BK1-BK6**

INSTRUCTIONS TO THE STUDENTS:

This test book consists of THREE (3) sections:

Section A: True/False with Reasoning	[10 Marks]
Section B: Short Structure	[65 Marks]
Section C: Programming Structure	[25 Marks]

ANSWER ALL QUESTIONS IN THE ANSWER BOOKLET PROVIDED.

(Please write down your section and your lecturer's name on your answer booklet)

Name	
I/C No.	
Year/Course	
Section	
Lecturer Name	

This examination book consists of **22** printed pages excluding this page.

SECTION A – TRUE/FALSE QUESTIONS [Total 10 marks]

There are **FIVE (5)** 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 : _____

Reason : _____

_____

_____

_____

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 : _____

Reason : _____

_____

_____

_____

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

Answer : _____

Reason : _____

_____

_____

_____

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 : _____

Reason : _____

_____

_____

_____

_____

5. The statement `int arr[size];` in the following C++ code in Figure 5 produces an error. This is due to the size of array cannot be initialized dynamically. Thus, constant expression is required.

```
int main()
{
    int size, i;
    cin>>size;
    int arr[size];
    for(i=1; i<=size; i++)
    {
        cin>> arr[i];
        cout <<"You entered : "<< arr[i];
    }
    system("pause");
    return 0;
}
```

Figure 5

Answer : _____

Reason : _____

_____

_____

_____

SECTION B – STRUCTURE QUESTIONS [Total marks 65]

There are **FIVE (5)** 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 6, complete the missing C++ (Figure 7) 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 6

```
#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

     _____________________________________________________________

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

Figure 7

(b) Consider the following output screen in Figure 8 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 8

Answer 1(b):

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 9

(a) Based on C++ code in Figure 9, 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):

_____

_____

_____

_____

_____

_____

_____

_____

Answer 2(a):

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):

Answer 3(a):

(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):

Answer 3(b):

4. (a) Based on the Program in Figure 10, 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 10

Answer 4(a):

_____

_____

_____

(b) What is the output of the above program (Figure 11) 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 11

Answer 4(b):

_____

_____

_____

_____

_____

_____

_____

_____

(c) Given the following excerpted program (Figure 12):

```
#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 12

(i) Based on the **`main()`** function in Figure 12, 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):

_____

_____

_____

_____

_____

_____

_____

_____

Answer 4(c-i):

Answer 4(c-i):

5. (a) Write a C++ function named **`ArrayEq`** that has 3 parameters: two arrays of integers, and their size (the same size for both arrays). The function should return `true` if and only if the arrays contain the same values in the same order. [5 marks]

Answer 5(a):

(b) Based on a C++ code in Figure 13, answer the following questions.

```
#include<iostream>
#include<cmath>
using namespace std;
#define M 4

int main()
{
  double a[M];
  int i;
  for (i=0;i<M;++i)
          a[i]=pow(i,2.0);

   for (i=0;i<M;++i)
           cout<<"["<<i<<"]"<<"="<<a[i]*2<<"\t";
           cout<<endl;

   system("pause");
   return 0;
}
```

Figure 13

(i) What is the output of the above program? [4 marks]

Answer 5(b-i):

_____

_____

_____

_____

_____

_____

_____

_____

(ii) Extended from C++ code in Figure 13, write a function named **`calculate`** that will calculate and print the square root of the sum of all elements in array `a[]`. [5 marks]

Answer 5(b-ii):

(iii) Write a statement in `main()` that will invoke function `calculate` constructed in (b-ii). [1 mark]

Answer 5(b-iii):

_____

_____

_____

SECTION C – PROGRAMMING QUESTION [Total marks 25]

Section C consists of **ONE (1)** question only. You are required to answer this question in the answer space provided.

Every month, UTM-Mobile will summarize the profit of their telecommunication service. Once a month, the staff of UTM-Mobile will record the telecommunication charges for every customer. Write **C++ program** based on the following requirements:

(i) The program consists of **`main()`** program and TWO (2) user-defined functions as listed in Table 4.

(ii) In **`main()`** program, the user will input a list of customer's detail as depicted in Figure 14. The number of customer is depended on the user's input.

(ii) Next, the **`main()`** program will call function **`generateOverdue()`** together with sending parameters as described in Table 4 to acquire the total overdue amount.

(ii) Then, the **`main()`** program will call function **`summaryProfit()`** together with sending parameters as described in Table 4 in order to display the profit summary of their telecommunication service onto the output screen as described in Figure 15.

(iv) For any floating value, it should be formatted into TWO (2) decimal places.

The input to the program might be as follows:

```
Enter number of customer: 10
Customer #1
Name: XXX
Account number: XXX
To-date charges (RM): XXX
To-date payment (RM):  XXX
…
…
…
Customer #10
Name: XXX
Account number: XXX
To-date charges (RM): XXX
To-date payment (RM): XXX
```

Figure 14

The format of the output might be as follows:

```
UTM-Mobile MONTHLY PROFIT
Number of customer: 10
Total Charges (RM): XX.XX
Total Paid (RM): XX.XX
Total Overdue (RM): XX.XX
```

Figure 15

The description of user-defined functions needed as follows:

Table 4. User-defined functions

Function	Description
`generateOverdue`	This function receives the following parameters: • Number of customer • A list of customer's name • A list of customer's account number • A list of customer's charges • A list of customer's payment This function calculates the overdue amount for affected customers only. The formula is given as follows: `overdue = paid - charges` Next, the function will sum up the overdue amount and returns that total to the `main()` program. Besides, the summary of overdue amount for every customer will be written into output file named as "**overdue.txt**". The example of output file is described in Figure 16.

Function	Description
`summaryProfit`	This function receives the following parameters: • Number of customer • Total overdue • A list of customer's charges • A list of customer's payment This function calculates the total of customer's charges and customer's charges payment. Next, the function will display the profit summary as depicted in Figure 15.

The format of the output file is as follows:

Figure 16

Answer 5:

Answer 5:

Answer 5:

Answer 5:

Answer 5: