UNIVERSITI TEKNOLOGI MALAYSIA

FINAL EXAMINATION
PAPER I

SEMESTER I 2014/2015

SUBJECT CODE	**: SCJ1013 / SCSJ1013**
SUBJECT NAME	**: PROGRAMMING TECHNIQUE I (PAPER 1)**
YEAR/COURSE	**: 1 (SCSJ / SCSV / SCSB / SCSR)**
TIME	**: 9.00 A.M. – 11.30 A.M. (2 HOURS AND 30 MINUTES)**
DATE	**: 4 JANUARY 4, 2015**
VENUE	**: KOLEJ TUN DR ISMAIL (KTDI)**

INSTRUCTIONS TO THE STUDENTS:

This test book consists of 9 structured questions.

TOTAL **[70marks]**

ANSWER ALL QUESTIONS IN THIS BOOKLET IN THE SPACES PROVIDED.

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

This question booklet consists of 14 pages including the cover page.

1. **Program 1** given below is partially completed. Complete the program by filling in the blank lines according to the comments given for each respective line. **(5 marks)**

```
//Program 1
#include <iostream>
____________________//(i)use an appropriate library for file operations

using namespace std;

int main()
{
    ______________; //(ii) declare a file variable for an input file
    ______________; //(iii) declare a file variable for an output file

     int score;     // to store the number read from the input file
     int total = 0; // the total score

     ____________________; //(iv) open an input file named numbers.txt
     ____________________; //(v) open an output file named totScore.tx

     // Reading a list of numbers from the input file
     while ( _______________ ) //(vi) read input until the end of file
     {     ________________;   //(vii) read score from the input file
          total += score;
     }

     cout<< "Total score is " << total <<endl;
     ___________________<< endl; //(viii) print the total score into
                                 //         the output file.

     ________________;     //(ix) close the input file
     ________________;     //(x) close the output file

     cout << "Done!" << endl;

     return 0;
}
```

2. **`Program 2 is`** given below. What is the output produced by this program? **(6 marks)**

```
// Program 2
#include <iostream>
using namespace std;

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

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

int sum(int a, double b) {
      return a+b;
}

void copy (int& a, int& b){
      a *= 2;
      b *= 2;
      cout << "a =" << a << ", b =" << b << endl;
}

int main(){
      int x = 2, y =3 , z=0;
      sum(x, x);
      z = sum(y,1);
      cout << "x =" << x << " , z =" << z << endl;
      copy (x, y);
      sum(x,y,z);
      cout << "x =" << x << ", y =" << y << ", z =" << z;
      return 0;
}
```

3. Given `Program 3` as follows. What is the output of this program? **(3 marks)**

```
// Program 3
#include <iostream>
using namespace std;

int z = 0;

int f (int k, int m)
{
     static int n = 10;
     n = n+k;
     m = n;
     return k+n;
}

int main()
{
     int k = 3, m = 5;
     cout << k << " " << m <<  " " << f(k,m) << endl;

     z = f(k,m);
     cout << k << " " << m <<  " " << z << endl;

     return 0;
}
```

4. A length can be measured in miles, or yards and feet. A mile is equivalent to 5280 feet and a yard is equivalent to 3 feet.

   a) Write a function named **`yardToMile`** to convert measurements in **`yards`** and **`feet`** to measurements in **`miles`**. The function accepts input of **`yards`** and **`feet`** (whole numbers) and returns an output of the total number of **`miles`** (a floating-point value). Use appropriate data types, parameter passing and return mechanisms in creating your function. Assume that this function is called from the main function with different sets of values and prints the output in **`miles`**.

   (5 marks)

   b) Then, write a call statement to invoke the function defined in (a) to convert the length of 8 yards and 2 feet into miles. **(1 mark)**

5. Given a two-dimensional array as declared in the **`Program 5`** below. Based on the program, answer the questions following it.

```
//Program 5
#include <iostream>
#include <cmath>
using namespace std;

#define NROW 2
#define NCOL 3

int main(){
      int numbers[NROW][NCOL];

      for (int i=0; i<NROW; ++i)
            for (int j=0; j<NCOL; ++j)
                  numbers[i][j] = pow((i+j),2.0);

      for (int i=0; i<NROW; ++i){
            for (int j=0; j<NCOL; ++j)
                  cout << numbers[i][j]*2 << "\t";
            cout << endl;     }

      return 0;
}
```

a) Write the output printed by the program. **(3 marks)**

b) Define a function named **`calculate`** to calculate and print the square root of the sum of all the elements in an array. The function should accept a one-dimensional array as its parameters. **(5 marks)**

c) Then, in the main function, write a statement that will invoke the function **`calculate`** to calculate and print the square root of the sum of all the elements for each row in the array **`numbers`.**

(1 mark)

6. **`Program 6`** below is incomplete and it does not include the definitions of functions **`inputData`**, **`compare`** and **`compute`**. Complete the definition of each of these functions according to the requirements stated as comments in the next page. Calls to these functions are shown in the main function.

```
//Program 6
#include <iostream>
using namespace std;
#define SIZE 5

void inputData(int[]);
void compare(int [],int [] );
int compute(int [],int []);

int main()
{  int arr1[SIZE],sum=0;
   int arr2[SIZE]={2, 3, 2, 2, 1};

   inputData(arr1);
   compare(arr1,arr2);
   sum = compute(arr1,arr2);

   cout << "Sum of elements in new array :" << sum << endl;
   return 0;
}
```

// (a) Function **`inputData.`** This function accepts an array that is to be filled with integer values
// entered by the user. **(2 marks)**

____

____

____

____

____

// (b) Function **`compare.`** This function accepts two arrays. It will compare the content of both
// arrays for equality element-by-element. For example, if the first element of both arrays are
// equal, the function will display the message **"Equal"**, otherwise the message **"Not Equal"**
// will be displayed. **(3.5 marks)**

____

____

____

____

____

// (c) Function **`compute.`** This function accepts two arrays. It also declares another array to store
// the multiplication of each element for both arrays sent to this function. For example, if
// the sent arrays are **`arr1`** and **`arr2`**, and the new declared array is **`arr3`**, then **`arr3[0]`**
// will store the result of **`arr1[0] x arr2[0].`** The function will return the sum of all
// the elements of **`arr3`**. **(5.5 marks)**

____

____

____

____

____

____

7. **Table 1** indicates how the record of a patient is stored. Use the information provided to answer the questions in parts (a), (b) and (c).

Table 1

Field	Field Name	Field Data Type
Patient's name	`name`	string or array of 30 characters
Patient's doctor	`doctor`	string or array of 20 characters
Patient's ward	`pWard`	A structure type named `Ward` containing 2 fields: • Any integer between 1 – 5 to indicate `floor` • A character F or M to indicate `gender` either a female or male ward, respectively.
Admission date	`admitDate`	A structure type named `Date` containing 3 fields: • Any integer between 1 – 31 to indicate the day of admission • Any integer between 1 – 12 to indicate the month of admission • Any integer to indicate the year of admission

a) Write C++ statements to declare all the data types required for storing the record of a patient. Name the data type for a patient's record as **`PatientRecord`**.

(5 marks)

b) Declare a variable called `pRec` to store records for 50 patients. Set the twentieth patient's name to "Robert Kwok" and the ward to a male ward located on the fifth floor. **(2 marks)**

c) Write the statements to perform each of the following tasks:

i. Read in from the keyboard the thirty-fifth patient's doctor and admission date. **(2 marks)**

ii. Find the total number of male patients and the total number of female patients found on the second floor. **(3 marks)**

iii. If the last patient's name is "Mohammed", print out the patient's doctor and his/her ward information. **(2 marks)**

8. Given the declaration for a structured data type named **`S_TYPE`** as follows.

```
     struct S_TYPE
        {   float  fixedBefore;
            CHOICE choice;
            U_TYPE flexible;
            float  fixedAfter;
         };
```

Declare further:

a) the data type **`CHOICE`** which is an enumerated data type whose members are **`ONE`** and **`TWO`**. **(1 mark)**

b) the data type **`U_TYPE`** which is a union data type which can be made up of either **`choice1`** a character data type or **`choice2`** an integer data type. **(1 mark)**

9. A structure type called **`SalesRecord`** is designed to hold all necessary data regarding a salesman in a company and the sales information that he or she has made. The **`SalesRecord`** structure and an array variable called **`salesRep`** are declared below:

```
struct SalesRecord
{
     int id;
     char name[30];
     double quarterlySales[4];
     double totalAnnual;
};

SalesRecord salesRep[20];
```

a) A function named **`readRecord`** is designed to read the information of all the salesmen of the company from the user. Assume that the company has 20 salesmen. Below is the prototype of the function.

```
void readRecord (SalesRecord []);
```

i) Write a complete definition for the function **`readRecord`** to read the salesman's id, name and quarterly sales. **(6 marks)**

ii) Write a proper call statement to the function **`readRecord`** to fill in the array **`salesRep.`** **(2 marks)**

b) The member **`totalAnnual`** of **`SalesRecord`** is used to hold the information about the total amount of yearly sales of each salesman respectively. The calculation of the total sale is accomplished by the following function:

```
double getAnnualSale (double quarterlySales[])
{   double total = 0.0;
    for (int i=0; i< 4; i++)
       total += quarterlySales[i];
     return total;
 }
```

Write a call statement to this function to calculate the annual sale for every salesman in the list. **(3 marks)**

c) A function named **`getSalesman`** is used to search for the record of a salesman using the salesman’s id from the salesmen list of record. The following program segment is written to call to this function. Define the function **`getSalesman`** in order to accomplish its purpose. **(3 marks)**

```
:
SalesRecord searchRecord:
int id;
:
cout << “Enter Salesman’s ID =>” << endl;
cin >> id;
searchRecord = getSalesman(salesRep, id);
:
:
```

-- THIS PAGE IS INTENTIONALLY LEFT BLANK FOR ANSWERS --