14

EASTERN UNIVERSITY, SRI LANKA
DEPARTMENT OF MATHEMATICS
FIRST EXAMINATION IN SCIENCE (2011/2012)
FIRST SEMESTER (Jan,2015)
CS103 - INTRODUCTION TO PROGRAM DESIGN AND PROGRAMMING

ANSWER ALL QUESTIONS

TIME: TWO HOURS

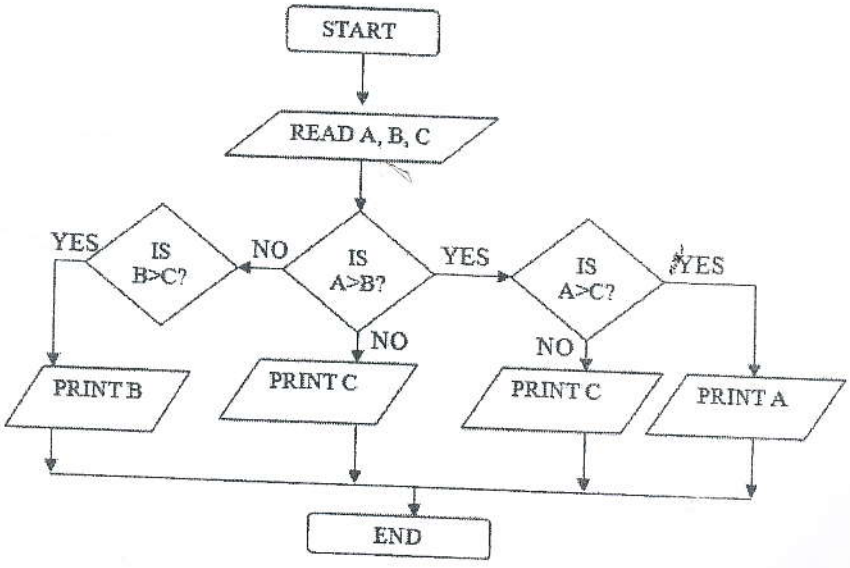
1)

a. Define *logical error* and *compile-time error* by giving examples for each error.

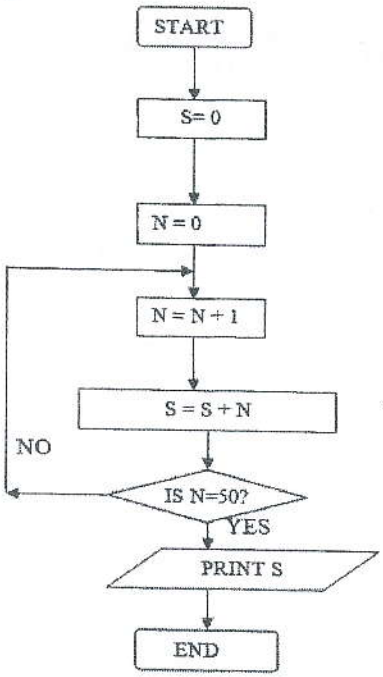
b. Define the following terms:
   I. Program;
   II. Algorithm;
   III. Flow chart;
   IV. Pseudocode.

c. Represent the following problems using "Flow chart":
   I. To determine the given letter is a "vowel" or a "non-vowel";
   II. To compute the factorial of a given number n (n!);
   III. To print the square of a given number n;
   IV. To find the roots of a quadratic equation $ax^2+bx+c=0$ where a is not equal to zero;
   V. To find and print the product of 3 numbers a, b and c.

d. Determine the problems specified by the following flow charts:
   I.



II.



2)

a. What do you mean by *Non-primitive* data type?

b. Explain the difference between *while loop* and *do-while loop* by giving suitabl examples.

c. Describe the break and continue statements by giving suitable examples.

d. Write the output of the following programs:

i.

```
#include<iostream.h>
void main()
{
    for( int i=0;i<8;i++)
    {
            if(i%2==0)
                    cout<<i+1<<"\t";
            else if(i%3==0)
                    cout<<i*i<<"\t";
            else if(i%5==0)
                    cout<<2*i-1<<"\t";
            else
                    cout<<i <<"\t";
    }
}
```

16

ii.
```
#include<iostream.h>
int main()
{
   int digit1,digit2,digit3;

   for(int i=150 ; i<=155 ; i++)
   {
          digit1=i/100;
          digit2=i/10 - digit1*10;
          digit3=i%10;

   if(digit1*digit1*digit1+digit2*digit2*digit2+ digit3*digit3*digit3 == i)
   cout<<i<<endl;
   }
   return 0;
}
```

e. Write a program to calculate the sum of first 10 natural numbers (use do-while loop).

f. Write a C++ program to print the following outputs:

I.

```
                A
            A   B
        A   B   C
    A   B   C   D
A   B   C   D   E
```

II.

```
1
212
32123
4321234
543212345
```

3)

a. Write a C++ function which accept two integers as an argument and return its sum.

```
int sum(int a, int b)
```

b. Write a C++ function to find the factorial of a positive integer n (n!).

```
int fact(int n)
```

c. Write a C++ function to swap values of two variables.

```
void swap(float a, float b)
```

d. Write a C++ function to count the number of words in a string.

```
void count(char S[])
```

e. Write a C++ function that returns x raised to the power n, where n can be any integer.

```
int power ( int x, int n)
```

4)

a. By giving suitable examples explain the parameter passing techniques: "*passing by value*" and "*passing by reference*"

b. Write the output for the following program:

I.

```
#include<iostream.h>
void main()
{
        int a=2,b=3,c;
        cout<<b--<<"\n";
        c=b*a--;
        cout<<c<<"\n";
        c+=c--*++a;
        cout<<c<<"\n";
        float d;
        d=b;
        cout<<d<<"\n";
        b*=++c%--a;
        cout<<b<<"\n";
}
```

II.

```
#include <iostream.h>
void main()
{
int v = 26;
int *p;          int **pTp;
p = &v;
pTp = &p;

cout << "  Value   = " << v << "\n";
cout << " *Pointer = " << *p << "\n";
cout << "**Pointer = " << **pTp << "\n";
v = 4805;
cout << "  Value   = " << v << "\n";
cout << " *Pointer = " << *p << "\n";
cout << "**Pointer = " << **pTp << "\n";
*p = -728;
cout << "  Value   = " << v << "\n";
cout << " *Pointer = " << *p << "\n";
cout << "**Pointer = " << **pTp << "\n";
**pTp = 945580;
cout << "  Value   = " << v << "\n";
cout << " *Pointer = " << *p << "\n";
cout << "**Pointer = " << **pTp << "\n";
}
```

c) Write a program to add two arrays A and B of size m x n.

d) Write a user defined function named Upper-half() which takes a two dimensional array A, with size N rows and N columns as argument and prints the upper half of the array.

Eg:
Input:

```
2 3 1 5 0
7 1 5 3 1
2 5 7 8 1
0 1 5 0 1
3 4 9 1 5
```

Output:

```
2 3 1 5 0
  1 5 3 1
    7 8 1
      0 1
        5
```

19

e) Declare a structure *Time* consisting of the following fields:
hours;
minutes;
seconds.

Write a function to convert the input into seconds:
Eg:
Input:

Hours:	1
Minutes:	10
Seconds:	1

Output:

Total seconds:	4201