C++ CASE STUDY

SCIENTIFIC CALCULATOR

BATCH NUMBER: 12

BATCH MEMBERS

S.NO	ROLL NUMBER	NAME	SIGNATURE
1	11	MOUKTHIK AVINASH	
2	12	VISHNU VARMA	
3	13	G THANMAY	
4	14	G ABHINAY	
5	15	G HANISHA	

CONTENTS:

CALCULATOR:

- A calculator is a device that performs arithmetic operations on numbers. The simplest calculators can do only addition, subtraction, multiplication, and division.
- More sophisticated calculators can handle exponential operations, roots, logarithms, trigonometric functions, and hyperbolic functions.
- Internally, some calculators actually perform all of these functions by repeated processes of addition.
- There are many types of calculator, scientific calculator is one of them

Scientific calculator:

- A scientific calculator is a type of electronic calculator, usually but not always handheld, designed to calculate problems in science, engineering, and mathematics.
- They have almost completely replaced slide rules in almost all traditional applications, and are widely used in both education and professional settings.
- In certain contexts such as higher education, scientific calculators have been superseded by graphing calculators, which offer a superset of scientific calculator functionality along with the ability to graph input data and write and store programs for the device. There is also some overlap with the financial calculator market.

Uses of scientific calculator:

Scientific calculators are used widely in any situation where quick access to certain mathematical functions is needed, especially those that were once looked up in tables, such as trigonometric functions or logarithms; they are also used in situations requiring calculations of very large or very small numbers, as in some aspects of astronomy, physics, and chemistry.

CONCEPTS:

1. CLASS
2. Objects
3. Scope resolution operator
4. Multilevel inheritance
5. Arrays and switch cases
6. Dynamic memory allocation
7. Header files

CLASSES:

In object-oriented programming languages like C++, the data and functions (procedures to manipulate the data) are bundled together as a self-contained unit called an object. A class is an extended concept similar to that of structure in C programming language; this class describes the data properties alone. In C++ programming language, class describes both the properties (data) and behaviors (functions) of objects. Classes are not objects, but they are used to instantiate objects.

What is a class?

- A class is an abstract data type similar to 'C'structure.
- Class representation of objects and the sets of operations that can be applied to such objects.
- Class consists of Data members and member functions.
- Private, Protected, Public are called visibility labels.
- The members that are declared private can be accessed by member functions only of that class.
- Public members can be accessed from outside the class also.
- In C++, data can be hidden by making it private.

Primary purpose of a class is to held data/information. This is achieved with attributes which is also known as data members.

The member functions determine the behavior of the class i.e. provide definition for supporting various operations on data held in form of an object.

Syntax:

```
class class_name
{
        ----------; //Data members
        ----------; //Member functions
};
```

Class Members:

- Data members and member functions are members of the class.
- Data Members and member functions must be declared within the class definition
- A member cannot be re-declared within a class.
- No member can be added elsewhere other than in the class definition.

Example:

```
class A
{
        int i; //Data member i
        int j; //Data member j
        void fun1(int, int); //Member function fun1
        void fun2(int, int); //Member function fun2
};
```

OBJECTS:

Class is mere a blueprint or a template. No storage is assigned when we define a class. Objects are instances of class, which holds the data variables declared in class and the member functions work on these class objects.

Each object has different data variables. Objects are initialized using special class functions called Constructors. We will study about constructors later.

And whenever the object is out of its scope, another special class member function called Destructor is called, to release the memory reserved by the object. C++ doesn't have Automatic Garbage Collector like in JAVA, in C++ Destructor performs this task.

Syntax:

```
class ABC
{
        int x; //Data member x
        void display(){}; //Empty member function
};
int main()
{
        ABC obj; //Declaration op object of class ABC
        obj.display();
        return 0;
}
```

Scope resolution operator:

C++ supports a mechanism to access a global variable from a function in which a local variable is defined with the same name as a global variable. It is achieved using the scope resolution operator. The syntax of accessing a global variable using scope resolution operator is as shown below.

Syntax:

:: global variable-name

The global variable-name to be accessed must be preceded by the scope resolution operator. It directs the compiler to access a global variable, instead of one defined as a local variable. Thus, the scope resolution operator permits a program to reference an identifier in the global scope that has been hidden by another identifier with the same name in the local scope.

Example:

```
#include  <iostream>
using namespace std;
int num=100;  //global variable
int main()
{
        int num=10;  //local variable
        cout<<"Local = "<<num; //accessing local variable
        cout<<"Global="<<::num; //accessing global variable
        return 0;
}
```

Output:

```
Local=10
Global=100
```

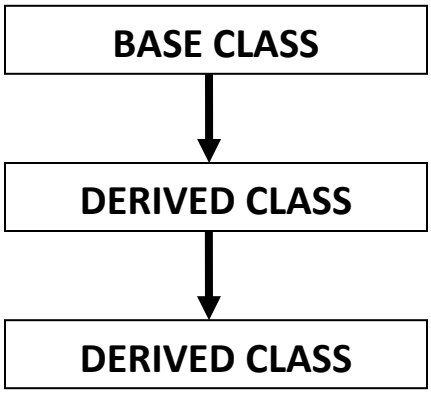
In the above example the variable num is declared both locally and globally. The notation "::" num refers to global variable.

INHERITANCE:

It is the process by which object of one class acquire the properties or features of objects of another class. The concept of inheritance provides the idea of reusability means we can add additional features to an existing class without modifying it. This is possible by driving a new class from the existing one. The new class will have the combined features of both the classes.

MULTILEVEL INHERITANCE:

In Multilevel Inheritance a derived class can also inherited by another class. For instance in the program below we have a base class and two derived classes. We are inheriting Base class in Derived1class and inheriting Derived1 class in Derived2class. So when a derived is inherited by another class then it creates Multiple Levels and this is knowan Multi-level Inheritance.



Syntax:

```
class  base_classname
{
        ----------; //Data members
        ----------; //Member functions
};
class intermediate_classname : visibility_mode base_classname
{
```

```
        properties;
        methods;
};
class child_classname : visibility_mode intermediate_classname
{
        properties;
        methods;
};
```

SWITCH CASE:

A switch case is used test variable equality for a list of values, where each value is a case. When the variable is equal to one of the cases, the statements following the case are executed. A break statement ends the switch case. The optional default case is for when the variable does not equal any of the cases.

Syntax:

```
switch(variable)
{
        case value-One:
                -------; //statements
                        break;
        casevalueTwo:
                -------; //statements
                        break;
        default:                //optional
                -------;//statements
}
```

The variable used in a switch statement can only be a short, byte, int or char. The values for each case must be the same data type as the variable type.

ARRAY:

- Arrays are kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.
- Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index.
- All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.

SYNTAX:

```
type array_Name[array_Size ];
```

DYNAMIC MEMORY ALLOCATION:

For dynamic memory allocation, C++ offers operator new. Operator new returns the pointer to the newly allocated space. If you want to allocate memory for one single element of a specified data type (it can be a built in data type, structure or class), you will have to use operator new in the following form:

```
new data_type;
```

If you want to allocate memory for an array, you will have to use another form of operator new:

```
variable=new data_type[size_of_array];
```

Dynamic memory allocation for two-dimensional arrays:

Often there is a need to allocate a memory for a two-dimensional array that is pointed up by a pointer to pointer. As you know from the “C++ Arrays”, two-dimensional array is an array of arrays.

A dynamic 2D array is a pointer to an array of pointers to arrays.If you want to allocate memory for a two-dimensional array dynamically, you have to create a pointer to pointer firstly:

```
int ** arr2D;
```

After this, you have to allocate memory for the array of pointers that will store the pointers to arrays:

```
arr = new int*[5];
```

Now, use a loop to allocate memory for each row of two-dimensional array:

```
for (int i = 0; i < 5; ++i)
        arr[i] = new int[3];
```

Operators delete:

Once you do not need memory allocated by operator new, you have to release it. You can do it by using operator delete:

delete pointer_variable; // for single pointer

delete []pointer_variable; //for an array of two pinters

In this case, a pointer is passed to the array as a parameter

```
delete [ ]arr;
```

Operator delete for 2D arrays:

If you want to free memory that was allocated for two-dimensional array dynamically, you have to free memory that was allocated for each row firstly. It can be done in a loop:

```
for (int i = 0; i < 3; ++i)
    delete[] arr[i];
```

After this, delete the pointer to the array of pointers:

```
delete[ ] arr;
```

When null pointer is passed to the operator delete, no effect will be produced.

HEADER FILES:

Header file	Description
iostream.h	This header file declares the basic C++ streams I/O routines.
conio.h	That's a (non standard) header that was used by some old MD-DOS C compilers to facilitate console input/output
math.h	This header file declares prototypes for the math functions and math error handlers. The routines in math.h file perform mathematical calculations and conversion.
#define	The #define creates a macro, which is the association of an identifier or parameterized identifier with a token string. After the macro is defined, the compiler can substitute the token string for each occurrence of the identifier in the source file

PROGRAM:

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
#define pi 3.14159265

class calculator
{
   public:
            char letter1,letter2,letter3,letter4,letter5,letter6;
            double a,b,n,c,k,result,*art;
            void geta()
            {
                    cout<<"\nEnter the value....";
                    cin>>a;
            }
            void getb()
            {
                    cout<<"\nEnter another value....";
                    cin>>b;
            }
};
```

```
class A: public calculator
{
   public:
   void AO();
   void TF();
   void LF();
   void PF();
   void IF();
};

void A::AO()
{
     cout<<"Arithematic operations :";
     cout<<"\n\n";
     cout<<"\t 1 : Addition \n";
     cout<<"\t 2 : Subtraction \n";
     cout<<"\t 3 : Multipilication \n";
     cout<<"\t 4 : Division \n\n";
     cout<<"Enter your choice....";
     letter1 = getch();
     //textattr(YELLOW);
     clrscr();
     cout<<"**************************************\n"<<endl<<endl;
```

```
switch(letter1)
{
        case '1':
                {
                    cout<<"Addition : "<<endl<<endl;
                    result=0;
                    cout<<"\n\nHow many nubers do you want to add : ";
                    cin>>n;
                    art=new double[n];
                    for(int i=0;i<n;i++)
                    {
                        cout<<"\nEnter the value of a"<<(i+1)<<" : ";
                        cin>>art[i];
                        result=result+art[i];
                    }
                    cout<<"\n\nResult = "<<result<<endl;
                    getch();
                    break;
                }
        case '2':
                {     cout<<"Subtraction :"<<endl<<endl;
                        geta();
                        getb();
                        result=a-b;
                        cout<<"\n\nResult = "<<result<<endl;
                        getch();
```

```
                break;
        }
case '3':
        {   cout<<"Multiplication :"<<endl<<endl;
                result=1;
                cout<<"\n\nHow many numbers do you want to multiply : ";
                cin>>n;
                art=new double[n];
                for(int i=0;i<n;i++)
                {
                        cout<<"\nEnter the value of a"<<(i+1)<<" : ";
                        cin>>art[i];
                        result=result*art[i];
                }
                cout<<"\n\nResult = "<<result<<endl;
                getch();
                break;
        }
case '4':
        {   cout<<"Division :"<<endl<<endl;
                geta();
                getb();
                result=a/b;
                cout<<"\n\nResult = "<<result<<endl;
                getch();
                break;
```

```
                        }
        }// end of inner switch
}// end of case 1 arithmatic operation
void A::TF()
{
    cout<<"\n\n";
    cout<<"\t1 : Sin function \n";
    cout<<"\t2 : Cos function \n";
    cout<<"\t3 : Tan function \n";
    cout<<"\t4 : Acos function\n";
    cout<<"\t5 : Asin function\n";
    cout<<"\t6 : Atan function\n\n";
    cout<<"Enter your choice....";
    letter2=getch();
    //textattr(YELLOW);
    clrscr();
    cout<<"*************************************\n"<<endl<<endl;
    switch(letter2)
    {
            case '1':
            {
                    cout<<"Sin function :"<<endl<<endl;
                    geta();
                    result=sin(pi*a/180);
                    cout<<"\n\nResult = "<<result<<endl;
                    getch();
```

```
        break;
}
case '2':
{
        cout<<"Cosine function :"<<endl<<endl;
        geta();
        result=cos(pi*a/180);
        cout<<"\n\nResult = "<<result<<endl;
        getch();
        break;
}
case '3':
{
         cout<<"Tangent function :"<<endl<<endl;
         geta();
         result=tan(pi*a/180);
         cout<<"\n\nResult = "<<result<<endl;
         getch();
         break;

}
case '4':
{
        cout<<"Arc cosine function :";
        geta();
        result=acos(a)*180/pi;
```

```
                cout<<"\n\nResult : "<<result<<endl;
                getch();
                break;
            }
            case '5':
            {
                cout<<"Arc sin function :";
                geta();
                result=asin(a)*180/pi;
                cout<<"\n\nResult : "<<result<<endl;
                getch();
                break;
            }
            case '6':
            {
                cout<<"Arc tangent function :";
                geta();
                result=atan(a)*180/pi;
                cout<<"\n\nResult : "<<result<<endl;
                getch();
                break;
            }
    }// inner switch
}//inner case 2 trignomatic
void A::LF()
{
```

```
cout<<"\n\n";
cout<<"\t1 : Natural log\n";
cout<<"\t2 : log with base 10 \n";
cout<<"\t3 : log with base as your choice \n\n";
cout<<"Enter your choice....";
letter3=getche();
//textattr(YELLOW);
clrscr();
cout<<"**************************************\n"<<endl<<endl;
switch(letter3)
{
        case '1':
        {       cout<<"Natural logarithmic function :"<<endl<<endl;
                geta();
                result=log(a);
                cout<<"\n\n Result = "<<result<<endl;
                getch();
                break;
        }
        case '2':
        {       cout<<"Logarithmic fuction with base 10 :"<<endl<<endl;
                geta();
                result= log10(a);
                cout<<"\n\n Result = "<<result<<endl;
                getch();
                break;
```

```
            }
        case '3':
        {     cout<<"Logarithmic fuction with base as your choice :"<<endl<<endl;
              geta();
              getb();
              result= log(a)/log(b);
              cout<<"\n\n Result = "<<result<<endl;
              getch();
              break;
        }
   }// end of switch
}// end of case 3 logrithmic
void A::PF()
{
   cout<<"\n\n";
   cout<<"\t1 : Power \n";
   cout<<"\t2 : Square root \n";
   cout<<"\t3 : Hypotenuse \n\n";
   cout<<"Enter your choice....";
   letter4=getche();
   //textattr(YELLOW);
   clrscr();
   cout<<"*************************************\n"<<endl<<endl;
   switch(letter4)
   {
        case '1':
```

```
{
        cout<<"Power function :"<<endl<<endl;
        geta();
        getb();
        result=pow(a,b);
        cout<<"\n\nResult = "<<result<<endl;
        getch();
        break;
}
case '2':
{     cout<<"Square function :"<<endl<<endl;
        geta();
        result=sqrt(a);
        cout<<"\n\nResult = "<<result<<endl;
        getch();
        break;
}
case '3':
{
        cout<<"Hypotenuse function :"<<endl<<endl;
        geta();
        getb();
        result=hypot(a,b);
        cout<<"Hypetenuse : "<<result;
        getch();
        break;
}
```

```
    }// end of switch
}// end of case power function
void A::IF()
{
    cout<<"\n\n";
    cout<<"\t1 : Integration \n";
    cout<<"\t2 : Differentiation \n\n";
    cout<<"Enter your choice....";
    letter6=getch();
    //textattr(YELLOW);
    clrscr();
    cout<<"**************************************\n"<<endl<<endl;
    switch(letter6)
    {
            case '1' : {
                    cout<<"Integration : ";
                    cout<<"\n\nEnter number of variables to integrate : ";
                    cin>>n;
                    result=0;
                    for(int i=0;i<n;i++)
                    {
                            cout<<"\nEnter the limits of variable x"<<i+1<<" : ";
                            cin>>a>>b;
                            cout<<"\nEnter coefficient and power of variable x"<<i+1<<" : ";
                            cin>>c>>k;
                            result=result+c*(pow(b,k+1)-pow(a,k+1))/(k+1);
```

```
                }
                cout<<"\nResult : "<<result<<endl;
                getch();
                break;
                }
        case '2' : {
                cout<<"Differentiation :";
                cout<<"\n\nEnter number of variables to differentiate : ";
                cin>>n;
                result=0;
                for(int i=0;i<n;i++)
                {
                        cout<<"\n\nEnter coefficient and power of variable x"<<i+1<<" : ";
                        cin>>c>>k;
                        cout<<"Enter the value of variable x"<<i+1<<" after differentiation :
";
                        cin>>a;
                        result=result+c*k*(pow(a,k-1));
                }
                         cout<<"\n\nResult : "<<result<<endl;
                         getch();
                        break;
                 }
    }
}
```

```
class cal : public A
{
   public:
   void choose();
};
void cal::choose()
{
   do
   {
   // textattr(GREEN);
    clrscr();

cout<<"*************************************************************\n*\t\t\t\t\t\t
\t\t*";
    cout<<"\n*\t 1 : Arithmetic Operations \t\t\t*\n*";
    cout<<"\t 2 : Trigonometric Functions \t\t\t*\n*";
    cout<<"\t 3 : Logarithmic Functions \t\t\t*\n*";
    cout<<"\t 4 : Power Functions \t\t\t\t*\n*";
    cout<<"\t 5 : Integration and Differentiation Function\t*\n*";
    cout<<"\t 6 : Exit...
   \t\t\t\t\t*\n*\t\t\t\t\t\t\t\t*\n*************************************************
   *********\n";
    letter5=getch();
    //textattr(3);
    clrscr();
    cout<<"*****************************************\n";
    switch(letter5)
```

```
    {
        case '1':
            AO();
            break;
        case '2':
            TF();
            break;
        case '3':
            LF();
            break;
        case '4':
            PF();
            break;
        case '5':
            IF();
            break;
    }
    }while(letter5!='6');
}
void star()
{
   cout<<"**************************************************************
****\n";
}
int main()
{
```

```
    //textattr(RED);
    clrscr();
    cal c;
    star();star();
    cout<<"********** SCIENTIFIC CALCULATOR
    **************************\n";
    star();star();
    getch();
    c.choose();
    return 0;
}
```

OUT PUT

```
**************************************************************************
**************************************************************************
********************** SCIENTIFIC CALCULATOR ***************************
**************************************************************************
**************************************************************************
```

```
************************************************************
*                                                          *
*        1 : Arithmetic Operations                         *
*        2 : Trigonometric Functions                       *
*        3 : Logarithmic Functions                         *
*        4 : Power Functions                               *
*        5 : Integration and Differentiation Function      *
*        6 : Exit...                                       *
*                                                          *
************************************************************
```

1)

```
*****************************************


        1 : Sin function
        2 : Cos function
        3 : Tan function
        4 : Acos function
        5 : Asin function
        6 : Atan function

Enter your choice....
```

2)

```
*****************************************


        1 : Natural log
        2 : log with base 10
        3 : log with base as your choice

Enter your choice....
```

3)

```
*****************************************


       1 : Power
       2 : Square root
       3 : Hypotenuse

Enter your choice...._
```

4)

```
*****************************************


       1 : Integration
       2 : Differentiation

Enter your choice....
```

5)

```
*****************************************
Arithematic operations :

        1 : Addition
        2 : Subtraction
        3 : Multipilication
        4 : Division

Enter your choice...._
```

1-1)

```
*****************************************


Addition :



How many nubers do you want to add : 3

Enter the value of a1 : 2

Enter the value of a2 : 43

Enter the value of a3 : 12


Result = 57
_
```

1-2)

```
****************************************

Subtraction :

Enter the value....65

Enter another value....23

Result = 42
```

1-3)

```
************************************

Multiplication :


How many numbers do you want to multiply : 3

Enter the value of a1 : 12

Enter the value of a2 : 5

Enter the value of a3 : 3

Result = 180
```

1-4)

```
****************************************

Division :

Enter the value....25

Enter another value....10

Result = 2.5
```

2-1)

```
****************************************

Sin function :

Enter the value....23

Result = 0.390731
_
```

2-2)

```
****************************************

Cosine function :

Enter the value....60

Result = 0.5
```

2-3)

```
****************************************

Tangent function :

Enter the value....45

Result = 1
```

2-4)

```
****************************************

Arc cosine function :
Enter the value....1

Result : 0
```

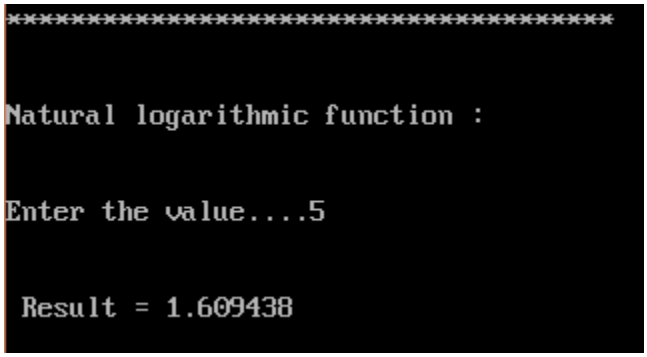
2-5)


****
Arc sin function :
Enter the value....0.5
Result : 30

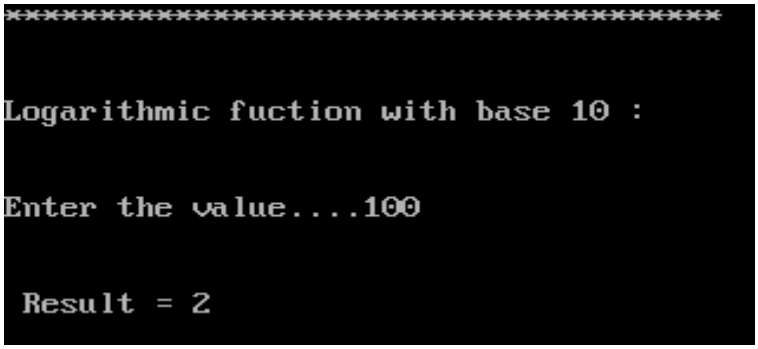
2-6)


****
Arc tangent function :
Enter the value....23
Result : 87.510447

3-1)


****
Natural logarithmic function :
Enter the value....5
Result = 1.609438

3-2)


****
Logarithmic fuction with base 10 :
Enter the value....100
Result = 2

3-3)

```
****************************************

Logarithmic fuction with base as your choice :

Enter the value....8

Enter another value....2

 Result = 3
```

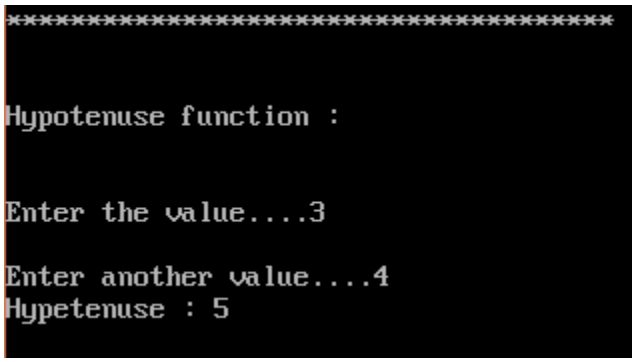
4-1)

```
********************************************

Power function :

Enter the value....3

Enter another value....4

Result = 81
```

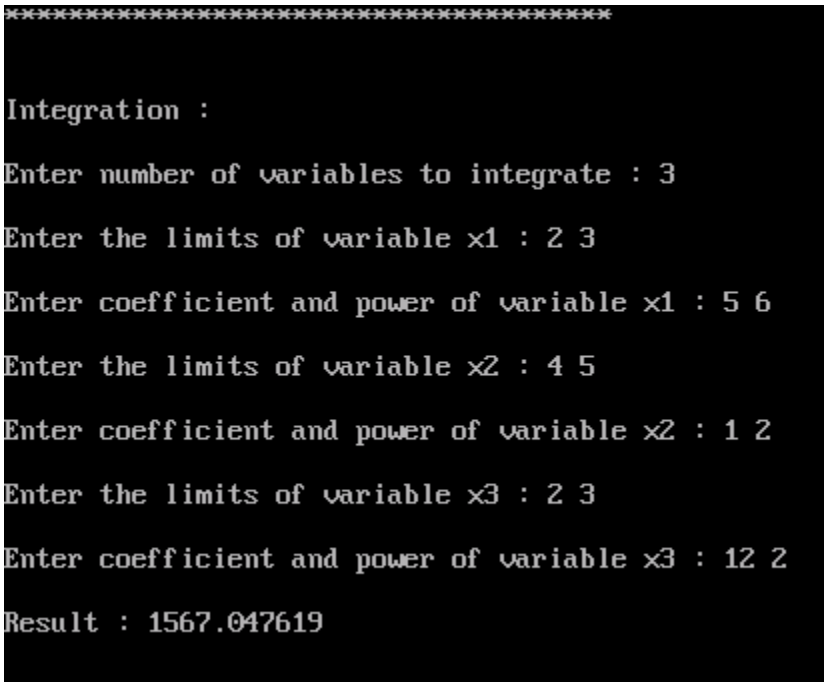
4-2)

```
********************************************

Square function :

Enter the value....256

Result = 16
```

4-3)


****
Hypotenuse function :
Enter the value....3
Enter another value....4
Hypetenuse : 5

5-1)


****
Integration :
Enter number of variables to integrate : 3
Enter the limits of variable x1 : 2 3
Enter coefficient and power of variable x1 : 5 6
Enter the limits of variable x2 : 4 5
Enter coefficient and power of variable x2 : 1 2
Enter the limits of variable x3 : 2 3
Enter coefficient and power of variable x3 : 12 2
Result : 1567.047619

5-2)

```
****************************************


Differentiation :

Enter number of variables to differentiate : 3


Enter coefficient and power of variable x1 : 2 3
Enter the value of variable x1 after differentiation : 3


Enter coefficient and power of variable x2 : 1 3
Enter the value of variable x2 after differentiation : 4


Enter coefficient and power of variable x3 : 3 4
Enter the value of variable x3 after differentiation : 5


Result : 1602
```

REFERENCES:

http://www.cplusplus.com/reference