1.CONTROL STATEMENTS USING CLASSES:

AIM: Program to demonstrate various control/loop statements using classes

APPARATUS:
1. code blocks IDE
2. ubuntu OS

SOURCE CODE:

```
#include <iostream>
    1.
        using namespace std;
        class forLoop
        {

        public:
          int length,coef=1,space,i,j;

           void print()
           {
              cout << "The output of for loop is:"<< endl;

            for(i=0;i<length;i++)
    {
        for(space=1;space<=length-i;space++)
        cout << "  ";
        for(j=0;j<=i;j++)
    2.          {
                    if (j==0||i==0)
                        coef=1;
                    else
                       coef=coef*(i-j+1)/j;
                    cout << "   " <<  coef;
                }
                cout << endl;
            }
        }

    };

    class whileLoop
    {
       public:

        int num;

void print()
       {
           cout << "The output of while loop is:";
           int factorial=1;
            cout << endl;

            int i=1;
            while(i<=num)
        {
            factorial=factorial*i;
             ++i;
        }

    cout<<"Factorial of Given Number is ="<<factorial<<endl;

        }

    };

    class doWhileLoop
    {
       public:

       int length;

       void print()
       {
          cout << "The output of do while loop is:";
```

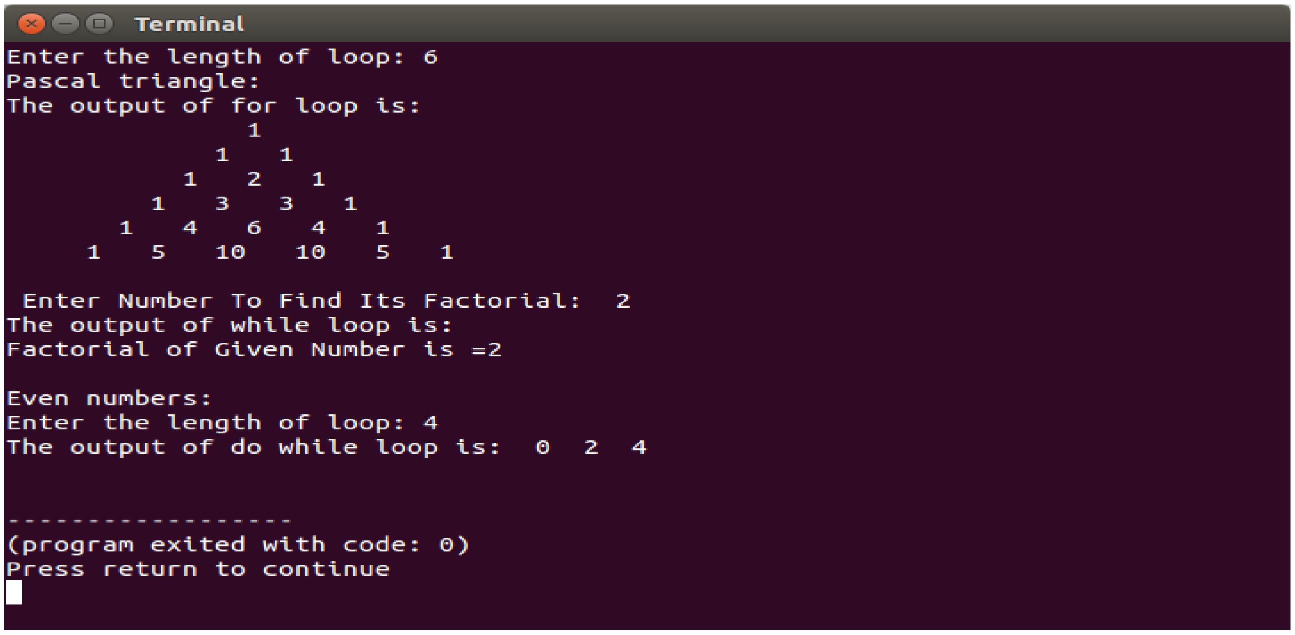
```
            int count_1=0;

            do
           {
             if(count_1%2==0)
             cout<<"  "<<count_1;
             count_1++;
           }
            while(count_1<=length);

         }

     };


int main()
     {
        int length;
        cout << "Enter the length of loop: ";
        cin >> length;
        cout << "Pascal triangle:" << endl;


        forLoop var1;
        var1.length=length;
        var1.print();
        cout << endl;

       int num;
        cout<<" Enter Number To Find Its Factorial:  ";
        cin >> num;
        whileLoop var2;
        var2.num = num;
        var2.print();
        cout << endl;


        cout<<"Even numbers: "<< endl;
        int length1;
        cout << "Enter the length of loop: ";
```

```
    cin >> length1;

    doWhileLoop var3;
    var3.length = length1;
    var3.print();
    cout << endl;

    return 0;
}
```

PROCEDURE:

1. Open code blocks IDE.
2. Create new project and select console application.
3. Enter the source code.
4. Check for errors.
5. Build and run the project.

OUTPUT:



```
Enter the length of loop: 6
Pascal triangle:
The output of for loop is:
               1
             1   1
           1   2   1
         1   3   3   1
       1   4   6   4   1
     1   5   10   10   5   1

 Enter Number To Find Its Factorial:  2
The output of while loop is:
Factorial of Given Number is =2

Even numbers:
Enter the length of loop: 4
The output of do while loop is:  0  2  4


------------------
(program exited with code: 0)
Press return to continue
```

RESULT:Hence,various control/loop statements are demonstrated using classes in code blocks IDE.

SOURCE:http://www.programiz.com/cpp-programming/examples/pyramid-pattern#pascal_triangle

http://fahad-cprogramming.blogspot.in/2013/02/program-to-find-factorial-in-C-Programming.html

2.MATRIX MULTIPLICATION:

AIM: Program to implement matrix multiplication

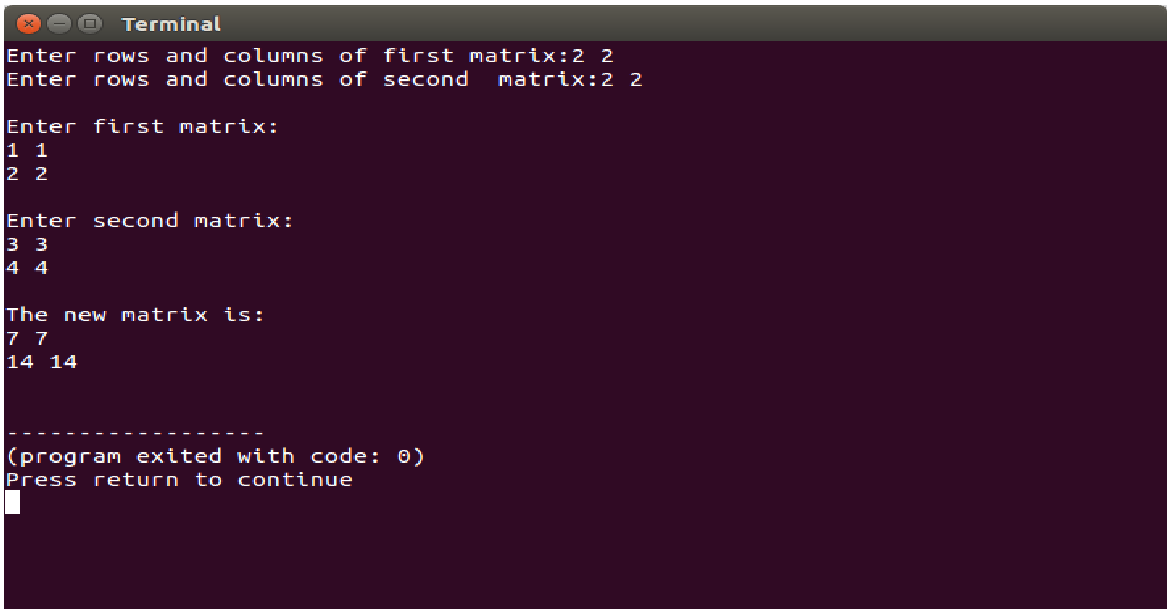
APPARATUS:
1. Code Blocks
2. Ubuntu OS

Source Code :
```
 #include<iostream>
 using namespace std;
 int main()
 {
     int a[5][5],b[5][5],c[5][5],m,n,p,q,i,j,k;
     cout<<"Enter rows and columns of first matrix:";
     cin>>m>>n;
     cout<<"Enter rows and columns of second  matrix:";
     cin>>p>>q;

     if(n==p)
     {
         cout<<"\nEnter first matrix:\n";
         for(i=0;i<m;++i)
             for(j=0;j<n;++j)
                 cin>>a[i][j];

         cout<<"\nEnter second matrix:\n";
         for(i=0;i<p;++i)
             for(j=0;j<q;++j)
                 cin>>b[i][j];

         cout<<"\nThe new matrix is:\n";
         for(i=0;i<m;++i)
         {
            for(j=0;j<q;++j)
            {
                c[i][j]=0;
                for(k=0;k<n;++k)
                   c[i][j]=c[i][j]+(a[i][k]*b[k][j]);
                 cout<<c[i][j]<<" ";
         }
```

```
            cout<<"\n";
         }
      }
      else
         cout<<"\nSorry!!!! Matrix multiplication can't be done";
      return 0;
 }
```

PROCEDURE:
1. Open code blocks IDE .
2. Create a new project and open console application.
3. Enter the source code
4. Check for errors.
5. Run and build the project.
6. Hence,ouput is obtained.

OUTPUT



```
Enter rows and columns of first matrix:2 2
Enter rows and columns of second  matrix:2 2

Enter first matrix:
1 1
2 2

Enter second matrix:
3 3
4 4

The new matrix is:
7 7
14 14


------------------
(program exited with code: 0)
Press return to continue
```

RESULT:
Hence, Program to implement matrix multiplication is executed.

SOURCE:
http://www.thecrazyprogrammer.com/2012/09/c-program-to-multiply-two-matrices.html

3.<u>STACKS USING ARRAYS</u>:

AIM: Program to implement Stack using Arrays.

APPARATUS:

1. Code Blocks IDE
2. Ubuntu OS

PROGRAM:

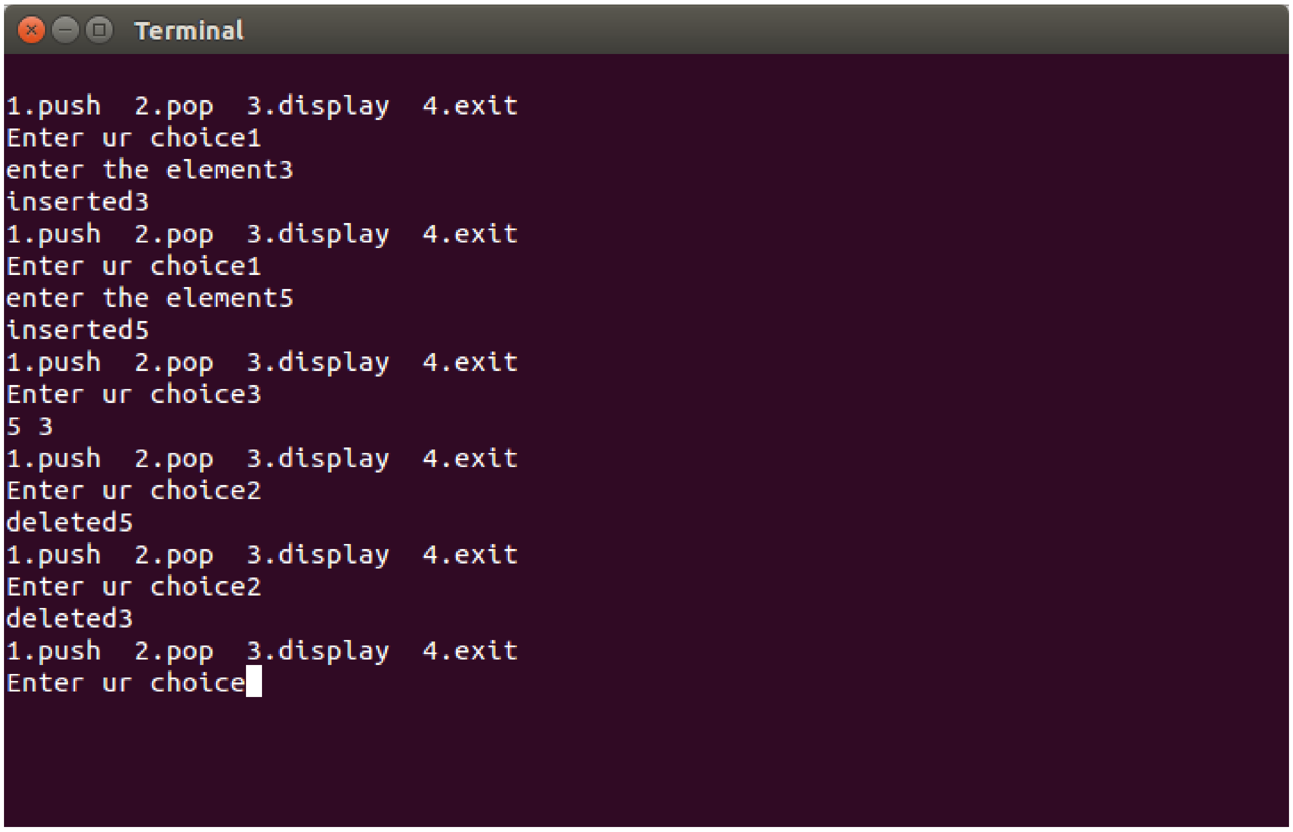
```
#include<iostream>
#include<stdlib.h>
using namespace std;
class stack
{
             int stk[5];
             int top;
      public:
             stack()
              {
                top=-1;
               }
             void push(int x)
              {
                 if(top >  4)
                        {
                            cout <<"stack over flow";
                            return;
                        }
                 stk[++top]=x;
                 cout <<"inserted" <<x;
               }
             void pop()
               {
                  if(top <0)
                   {
                         cout <<"stack under flow";
                         return;
                    }
                    cout <<"deleted" <<stk[top--];
                }
             void display()
               {
                   if(top<0)
```

```
                    {
                            cout <<" stack empty";
                            return;
                    }
                    for(int i=top;i>=0;i--)
                    cout <<stk[i] <<" ";
                }
                };

main()
{
     int ch;
     stack st;
     while(1)
        {
             cout <<"\n1.push  2.pop  3.display   4.exit\nEnter ur
choice";
             cin >> ch;
             switch(ch)
              {
               case 1:  cout <<"enter the element";
                        cin >> ch;
                        st.push(ch);
                        break;
               case 2:  st.pop();  break;
               case 3:  st.display();break;
               case 4:  exit(0);
               }
         }
return (0);}
```

PROCEDURE:

1.Open Code Blocks IDE.
2.Create a new project and open Console application.
3.Enter the source code.
4.Run and built the project.
5.Observe the output.

OUTPUT:



```
1.push  2.pop  3.display  4.exit
Enter ur choice1
enter the element3
inserted3
1.push  2.pop  3.display  4.exit
Enter ur choice1
enter the element5
inserted5
1.push  2.pop  3.display  4.exit
Enter ur choice3
5 3
1.push  2.pop  3.display  4.exit
Enter ur choice2
deleted5
1.push  2.pop  3.display  4.exit
Enter ur choice2
deleted3
1.push  2.pop  3.display  4.exit
Enter ur choice
```

RESULT: Hence, Program to implement stack using arrays is executed using code blocks.

SOURCE:
http://electrofriends.com/source-codes/software-programs/cpp-programs/cpp-data-structure/c-programs-to-implement-the-stack-adt-using-an-array/

4.<u>DEFAULT ARGUMENTS</u>:

AIM: Program to implement default arguments.

APPARATUS:

1.Code Blocks IDE
2.Ubuntu OS

PROGRAM:

```
#include<iostream>

using namespace std;

double volume1(double l=10,double b=20,double h=5);

int main()
{
	double length;
      double width;
      double height;
      double volume;

      cout<<"\n Enter the value of length = ";
      cin>>length;

      cout<<"\n Enter the value of width = ";
      cin>>width;

      cout<<"\n Enter the value of heigth = ";
      cin>>height;

      volume = volume1();
      cout<<"\n Volume with no argument passed =
"<<volume<<endl;

      volume=volume1(length);
      cout<<"\n Volume with one argument passed =
"<<volume<<endl;

      volume=volume1(length,width);
      cout<<"\n Volume with two argument passed =
"<<volume<<endl;
```

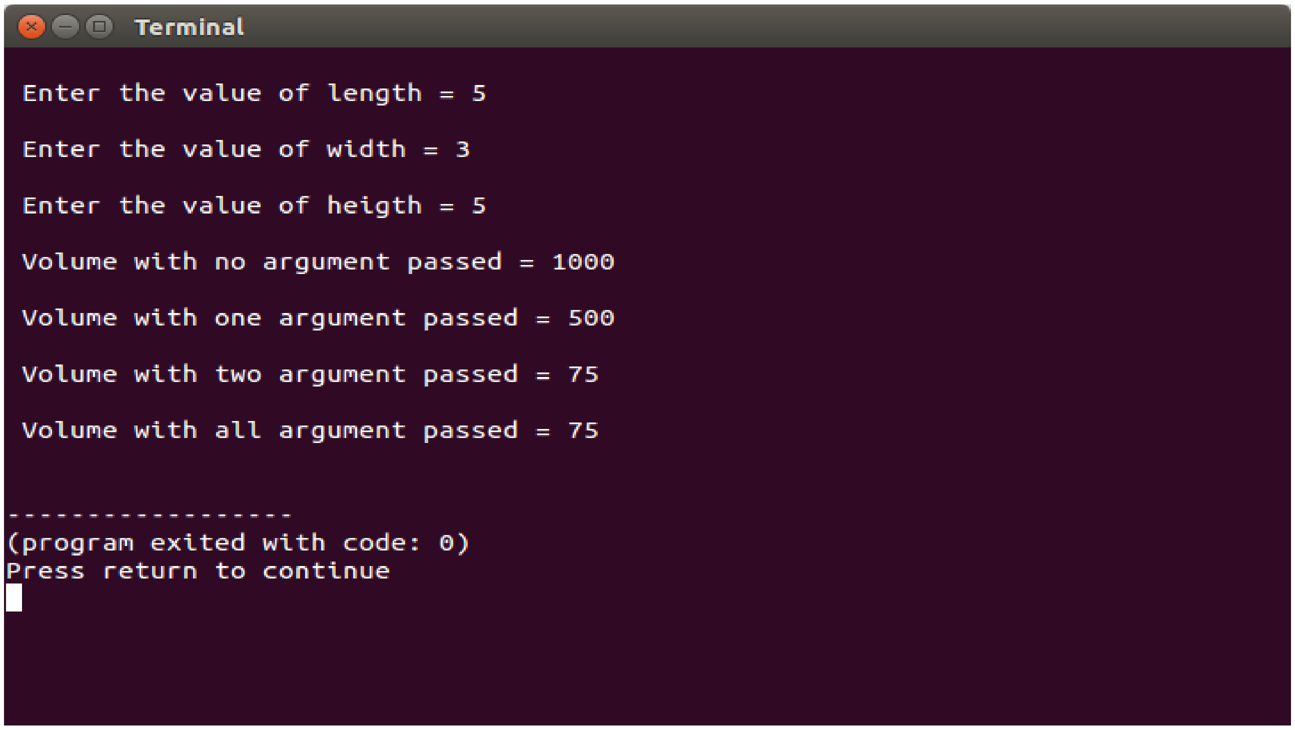
```
	volume=volume1(length,width,height);
	cout<<"\n Volume with all argument passed = 
"<<volume<<endl;

	return 0;
}

double volume1(double l,double b,double h)
{
	 return l*b*h;

}
```

PROCEDURE:
1.Open Code Blocks IDE.
2.Create a new project and open Console application.
3.Enter the source code.
4.Run and built the project.
5.Observe the output.

OUTPUT:

RESULT: Hence,Program for default arguments is executed.

SOURCE:
http://www.dailyfreecode.com/code/illustrate-default-argument-function-851.aspx\



```
 Enter the value of length = 5

 Enter the value of width = 3

 Enter the value of heigth = 5

 Volume with no argument passed = 1000

 Volume with one argument passed = 500

 Volume with two argument passed = 75

 Volume with all argument passed = 75


------------------
(program exited with code: 0)
Press return to continue
```

5.**FUNCTION OVERLOADING**:

AIM : program to implement function overloading

APPARATUS:
1.Code Blocks
2.Ubuntu OS

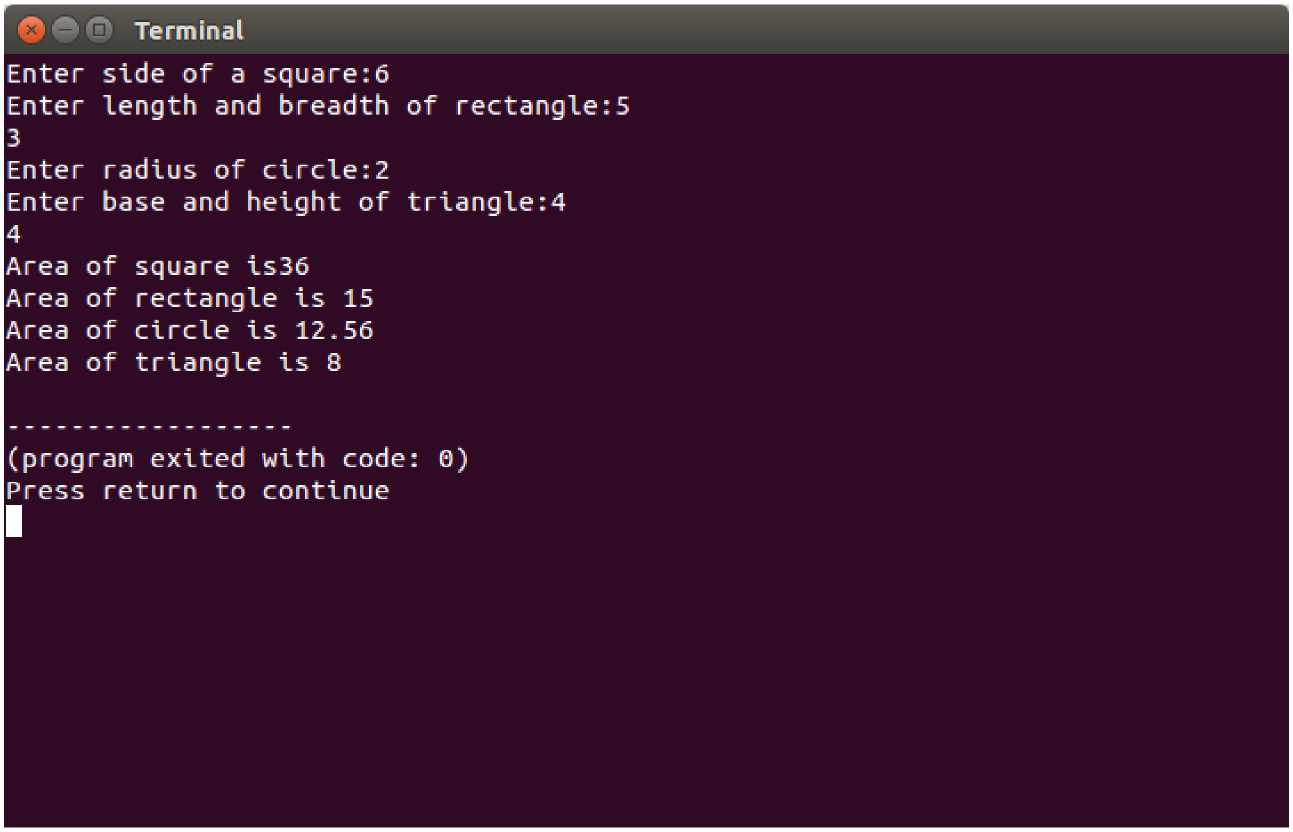
SOURCE CODE:

```
#include<iostream>
using namespace std;
int area(int);
int area(int,int);
float area(float);
float area(float,float);
int main()
{
        int s,l,b;
        float r,bs,ht;
        cout<<"Enter side of a square:";
        cin>>s;
        cout<<"Enter length and breadth of rectangle:";
        cin>>l>>b;
        cout<<"Enter radius of circle:";
        cin>>r;
        cout<<"Enter base and height of triangle:";
        cin>>bs>>ht;
        cout<<"Area of square is"<<area(s);
        cout<<"\nArea of rectangle is "<<area(l,b);
    cout<<"\nArea of circle is "<<area(r);
    cout<<"\nArea of triangle is "<<area(bs,ht);
}
int area(int s)
{
    return(s*s);
}
int area(int l,int b)
{
    return(l*b);
}
float area(float r)
{
    return(3.14*r*r);
}
float area(float bs,float ht)
{
   return((bs*ht)/2);
}
```

PROCEDURE:

1.open Code Blocks
2.create console application
3.enter the source code
4.check for errors
5.build and run the program
6.output is obtained.

OUTPUT:



```
Enter side of a square:6
Enter length and breadth of rectangle:5
3
Enter radius of circle:2
Enter base and height of triangle:4
4
Area of square is36
Area of rectangle is 15
Area of circle is 12.56
Area of triangle is 8

------------------
(program exited with code: 0)
Press return to continue
```

RESULT: Hence,program to implement function overloading is implemented.

SOURCE:
http://www.codequiz.in/c-program-to-find-area-of-squarerectanglecircle-and-triangle-by-using-function-overloading/

6.QUEUE USING ARRAYS:

AIM:To implement a Queue using arrays.

APPARATUS:

1.Code Blocks
2.Ubuntu OS

SOURCECODE:

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

class queue
{
              int queue1[5];
              int rear,front;
      public:
              queue()
                {
                     rear=-1;
                     front=-1;
                }
              void insert(int x)
               {
                   if(rear >  4)
                    {
                       cout <<"queue over flow";
                       front=rear=-1;
                       return;
                    }
                    queue1[++rear]=x;
                    cout <<"inserted" <<x;
               }
              void delet()
               {
                   if(front==rear)
```

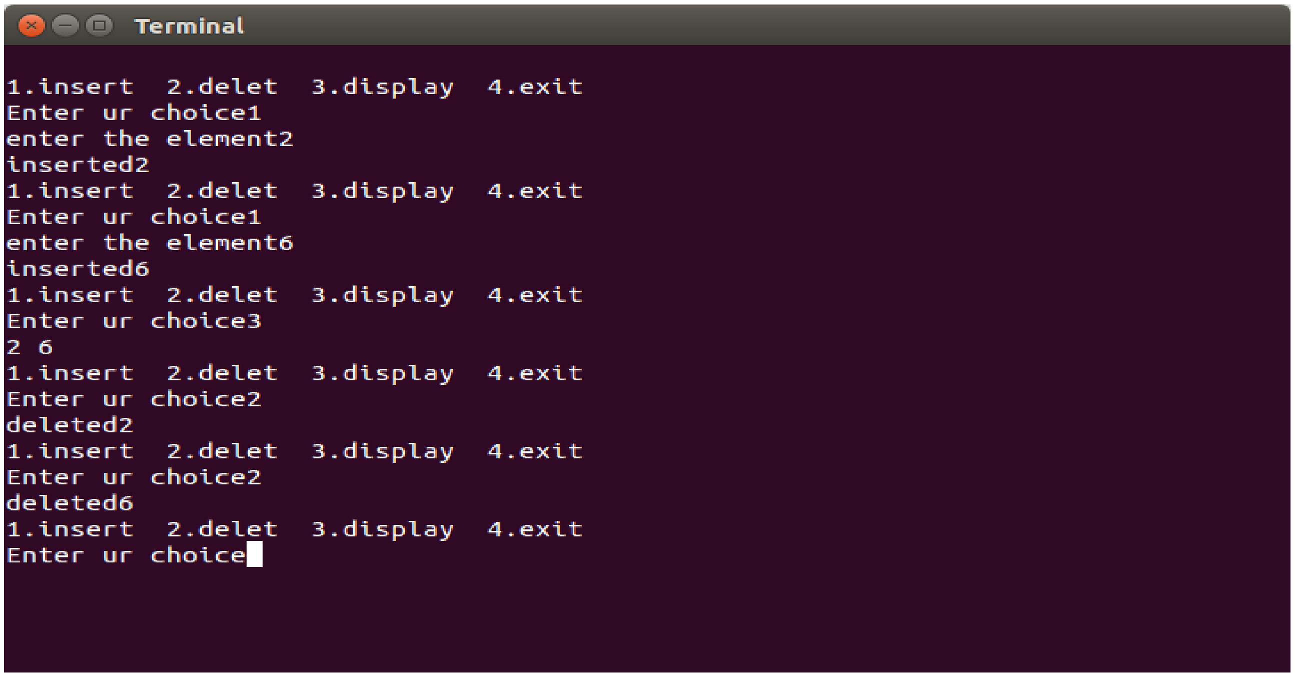
```
                    {
                        cout <<"queue under flow";
                        return;
                    }
                    cout <<"deleted" <<queue1[++front];
               }
             void display()
              {
                  if(rear==front)
                    {
                         cout <<" queue empty";
                         return;
                    }
                  for(int i=front+1;i<=rear;i++)
                  cout <<queue1[i]<<" ";
              }
};

int main()
{
      int ch;
      queue qu;
      while(1)
        {
             cout <<"\n1.insert  2.delet  3.display  4.exit\nEnter ur
choice";
             cin >> ch;
             switch(ch)
               {
                 case 1:    cout <<"enter the element";
                            cin >> ch;
                        qu.insert(ch);
                        break;
                 case 2:  qu.delet();  break;
                 case 3:  qu.display();break;
                 case 4: exit(0);
                 }
          }
return (0);
```

}

PROCEDURE:

1.open Code Blocks
2.create console application
3.enter the source code
4.check for errors
5.build and run the program
6.output is obtained.

OUTPUT:



```
1.insert  2.delet  3.display  4.exit
Enter ur choice1
enter the element2
inserted2
1.insert  2.delet  3.display  4.exit
Enter ur choice1
enter the element6
inserted6
1.insert  2.delet  3.display  4.exit
Enter ur choice3
2 6
1.insert  2.delet  3.display  4.exit
Enter ur choice2
deleted2
1.insert  2.delet  3.display  4.exit
Enter ur choice2
deleted6
1.insert  2.delet  3.display  4.exit
Enter ur choice
```

RESULT: Hence, Program to implement queue using arrays is executed.

SOURCE:http://electrofriends.com/source-codes/software-programs/cpp-programs/cpp-data-structure/c-programs-to-implement-the-queue-adt-using-an-array/

7.STATIC DATA MEMBERS:

AIM:program to implement static data members

APPARATUS:

1.Code Blocks
2.Ubuntu OS

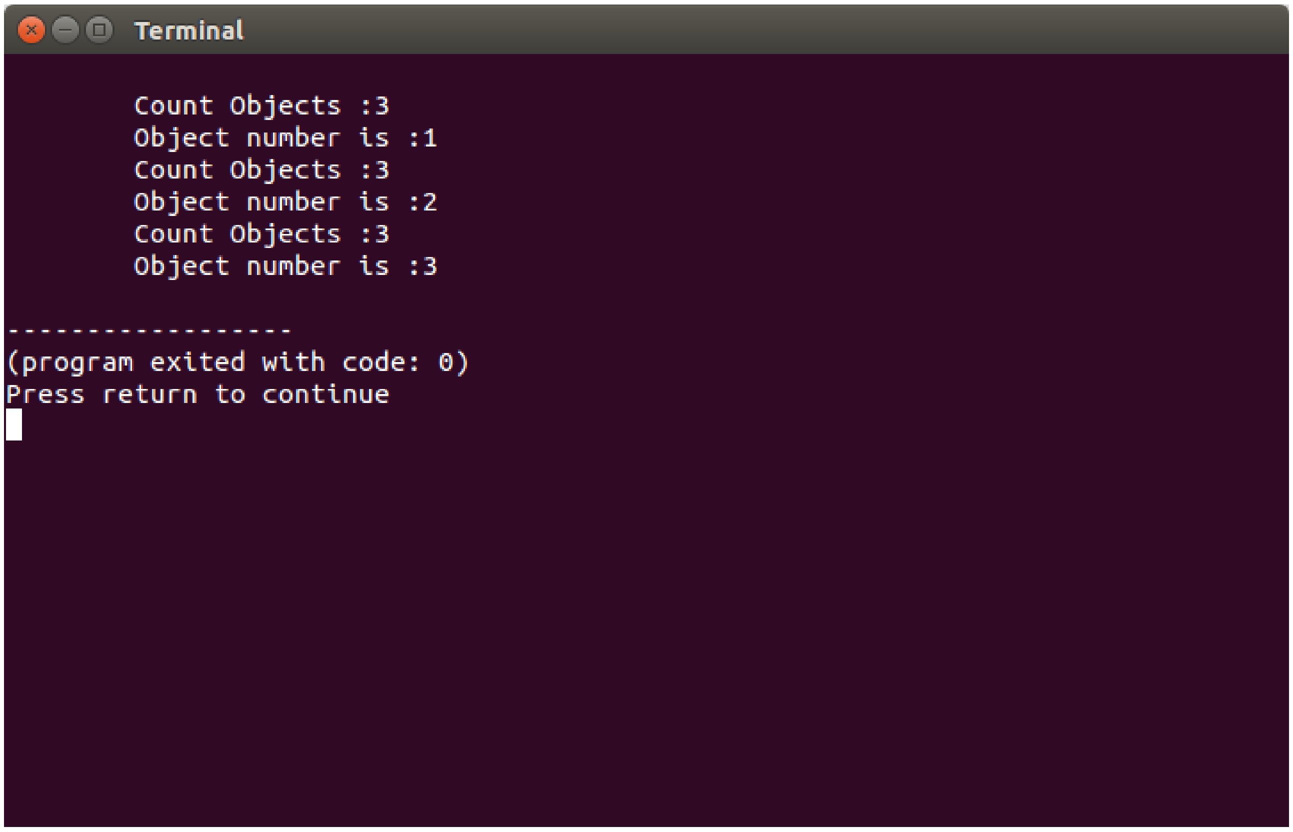
SOURCECODE:

```
#include<iostream>
using namespace std;
class stat
{
    int code;
    static int count;

   public:
    stat()
    {
      code=++count;
    }
    void showcode()
    {
      cout<<"\n\tObject number is :"<<code;
    }
    static void showcount()
    {
              cout<<"\n\tCount Objects :"<<count;
    }
};

int stat::count;
int main()
{
   stat obj1,obj2,obj3;

   obj1.showcount();
   obj1.showcode();
   obj2.showcount();
   obj2.showcode();
   obj3.showcount();
   obj3.showcode();
   return 0;
}
```

PROCEDURE:

1.open Code Blocks
2.create console application
3.enter the source code
4.check for errors
5.build and run the program
6.output is obtained.

OUTPUT:



```
        Count Objects :3
        Object number is :1
        Count Objects :3
        Object number is :2
        Count Objects :3
        Object number is :3

------------------
(program exited with code: 0)
Press return to continue
```

RESULT: Hence, Program to implement static data members is executed.

SOURCE:http://electrofriends.com/source-codes/software-programs/cpp-programs/cpp-advanced-programs/c-program-to-illustrate-the-static-member-function/

8.<u>ARRAY OF OBJECTS</u>:

AIM:program to implement array of objects

APPARATUS:

1.Code Blocks
2.Ubuntu OS

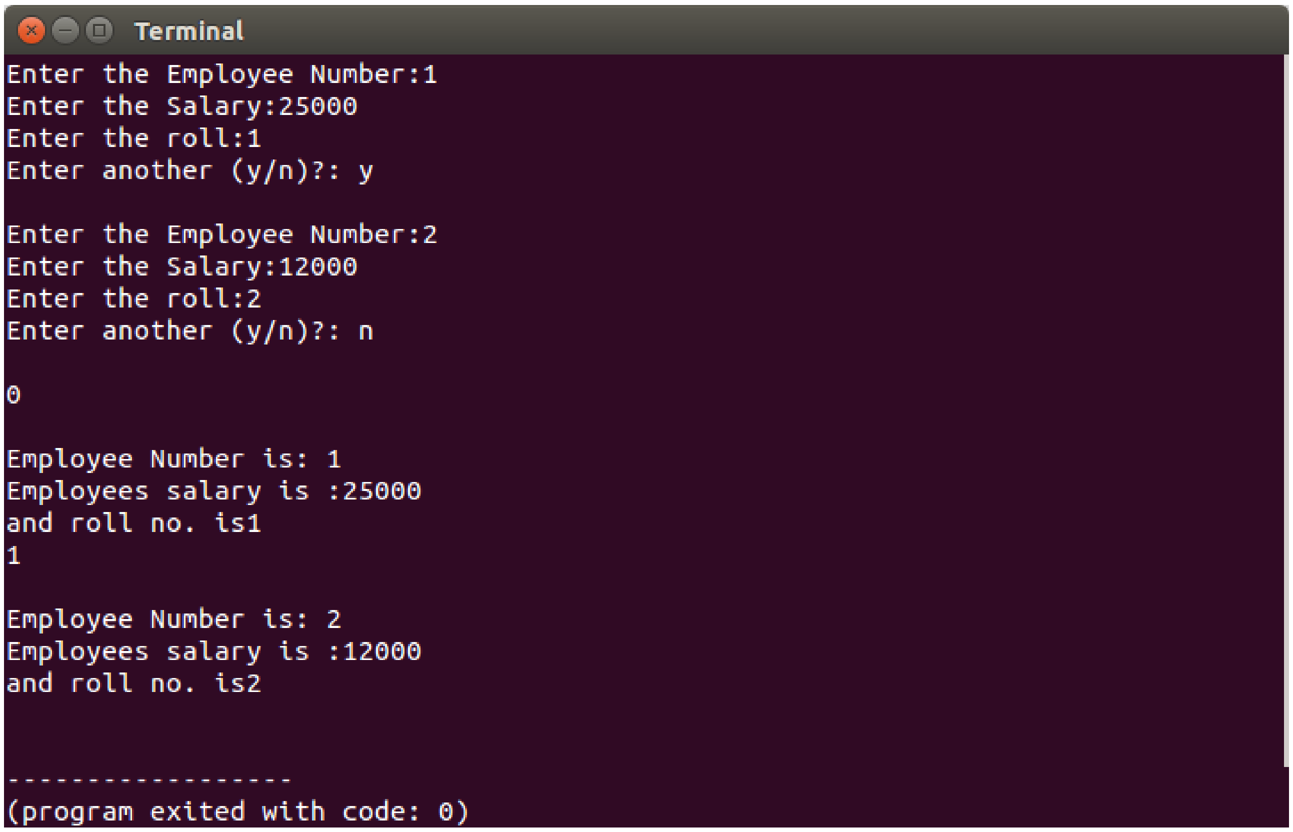
SOURCECODE:

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

class Details
{
private:
float salary;
float roll;
public:
void getname()
{
cout << "\nEnter the Salary:";
cin >> salary;
cout << "Enter the roll:";
cin >> roll;
}
void putname()
{
cout << "Employees salary is :" << salary<< endl <<"and roll
no. is" << roll << '\n';
}
};

int main()
{
Details det[50];
int n=0;
char ans;
do{
cout << "Enter the Employee Number:" << n+1;
det[n++].getname();
cout << "Enter another (y/n)?: " ;
cin >> ans;
cout << endl;
} while ( ans != 'n' );

for (int j=0; j<n; j++)
{
        cout<< j <<endl;
    cout << "\nEmployee Number is: " << j+1 << endl;
    det[j].putname();

}
 return 0;
}
```

PROCEDURE:

1.open Code Blocks
2.create console application
3.enter the source code
4.check for errors
5.build and run the program
6.output is obtained.

OUTPUT:



```
Enter the Employee Number:1
Enter the Salary:25000
Enter the roll:1
Enter another (y/n)?: y

Enter the Employee Number:2
Enter the Salary:12000
Enter the roll:2
Enter another (y/n)?: n

0

Employee Number is: 1
Employees salary is :25000
and roll no. is1
1

Employee Number is: 2
Employees salary is :12000
and roll no. is2


------------------
(program exited with code: 0)
```

RESULT: Hence, Program to implement array of objects is executed.

SOURCE:https://www.hscripts.com/tutorials/cpp/array-of-objects.php

9.<u>FRIEND FUNCTIONS</u>:

AIM: Implement a Program for friend functions

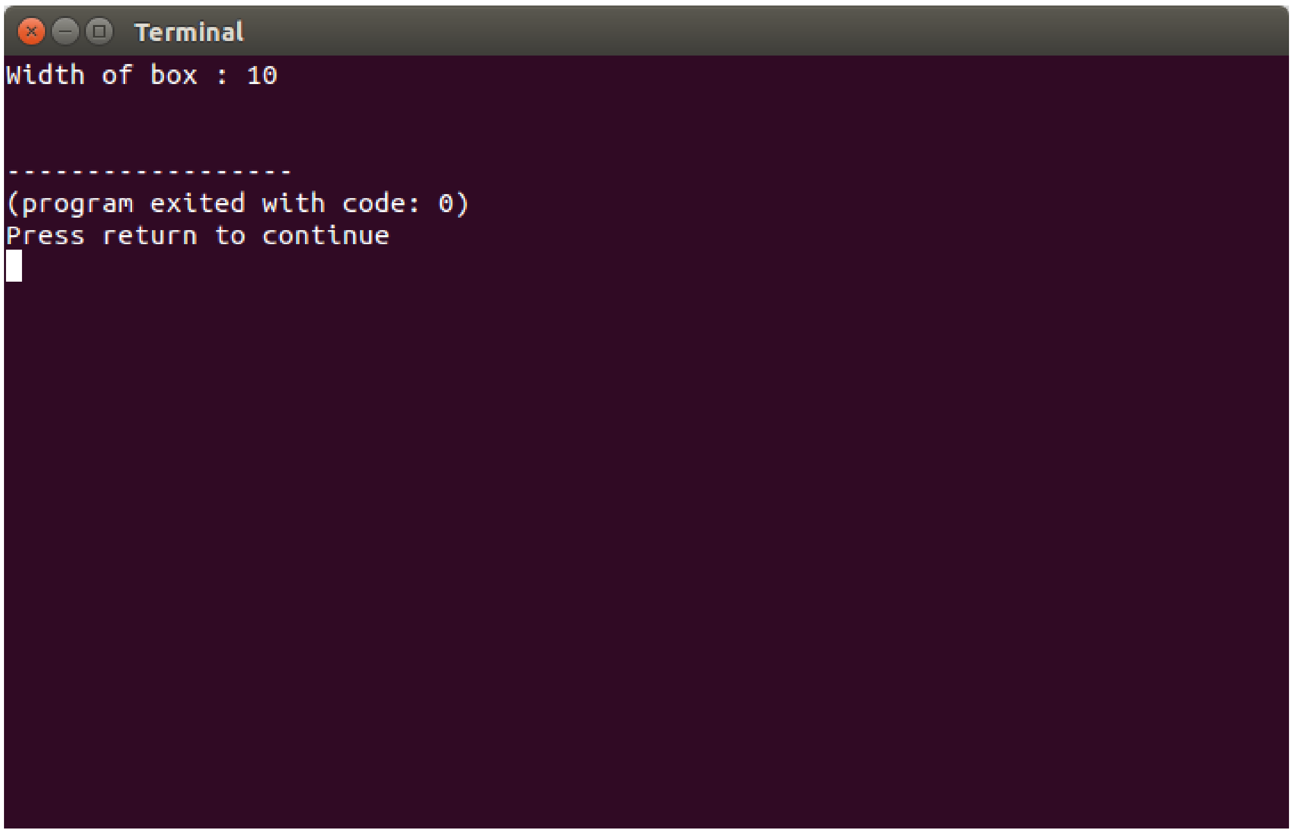
APPARATUS:
1)Ubuntu Os
2)code::blocks

SOURCE CODE:

```
#include <iostream>
using namespace std;
class Box
{
   double width;
public:
   friend void printWidth( Box box );
   void setWidth( double wid );
};
void Box::setWidth( double wid )
{
    width = wid;
}
void printWidth( Box box )
{

   cout << "Width of box : " << box.width <<endl;
}

int main( )
{
   Box box;

   box.setWidth(10.0);

   printWidth( box );
   return 0;
}
```

PROCEDURE:
1.open Code Blocks
2.create console application
3.enter the source code
4.check for errors
5.build and run the program
6.output is obtained.

OUTPUT:



```
Width of box : 10



------------------
(program exited with code: 0)
Press return to continue
```

RESULT: Hence the program for friend function is implemented.

SOURCE:
http://www.tutorialspoint.com/cplusplus/cpp_friend_functions.htm

10.CLASS CONSTRUCTOR:

AIM: Program to demonstrate constructor

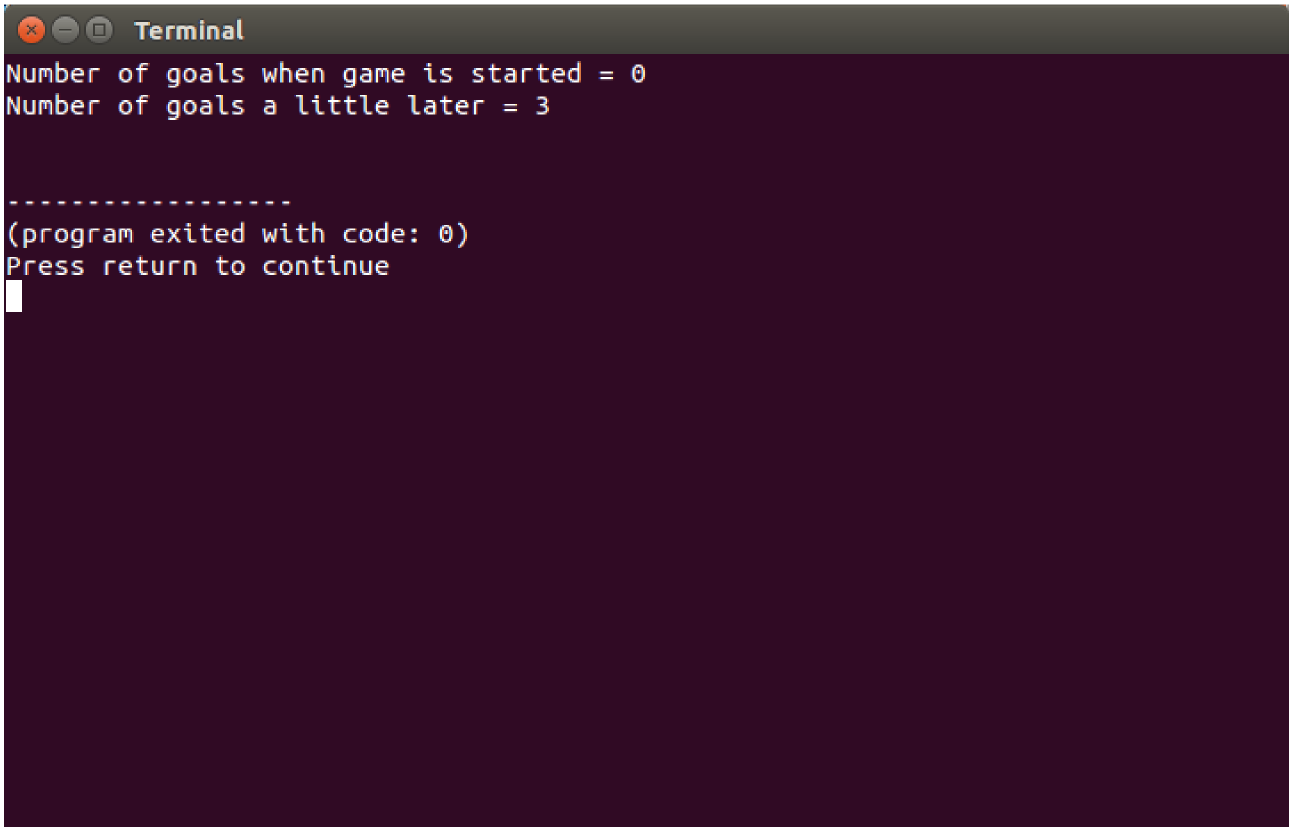
APPARATUS:
1.Code Blocks
2.Ubuntu OS

SOURCE CODE:

```
#include <iostream>
using namespace std;
class Game {
private:
   int goals;
public:
  // constructor used to initialize
  Game() {
    goals = 0;
  }

  // return score

  int getGoals() {
    return goals;
  }

  // increment goal by one

  void incrementGoal() {
    goals++;
  }
};

int main() {
  Game football;
  cout << "Number of goals when game is started = " <<
football.getGoals() << endl;

  football.incrementGoal();
  football.incrementGoal();
  football.incrementGoal();

  cout << "Number of goals a little later = " <<
football.getGoals() << endl;

  return 0;
}
```

PROCEDURE:
1.Open CodeBlocks
2.Create new project and open console application
3.Enter source code
4.Run and build Project
5.The output is obtained

OUTPUT:



```
Number of goals when game is started = 0
Number of goals a little later = 3


------------------
(program exited with code: 0)
Press return to continue
```

RESULT:
Hence, the program to demonstrate constructors is executed.

SOURCE:
http://www.programmingsimplified.com/cpp/source-code/constructor-program-example

11.SINGLE INHERITANCE:

AIM: Program to demonstrate single inheritance

APPARATUS:
1.Code Blocks
2.Ubuntu OS

SOURCE CODE:

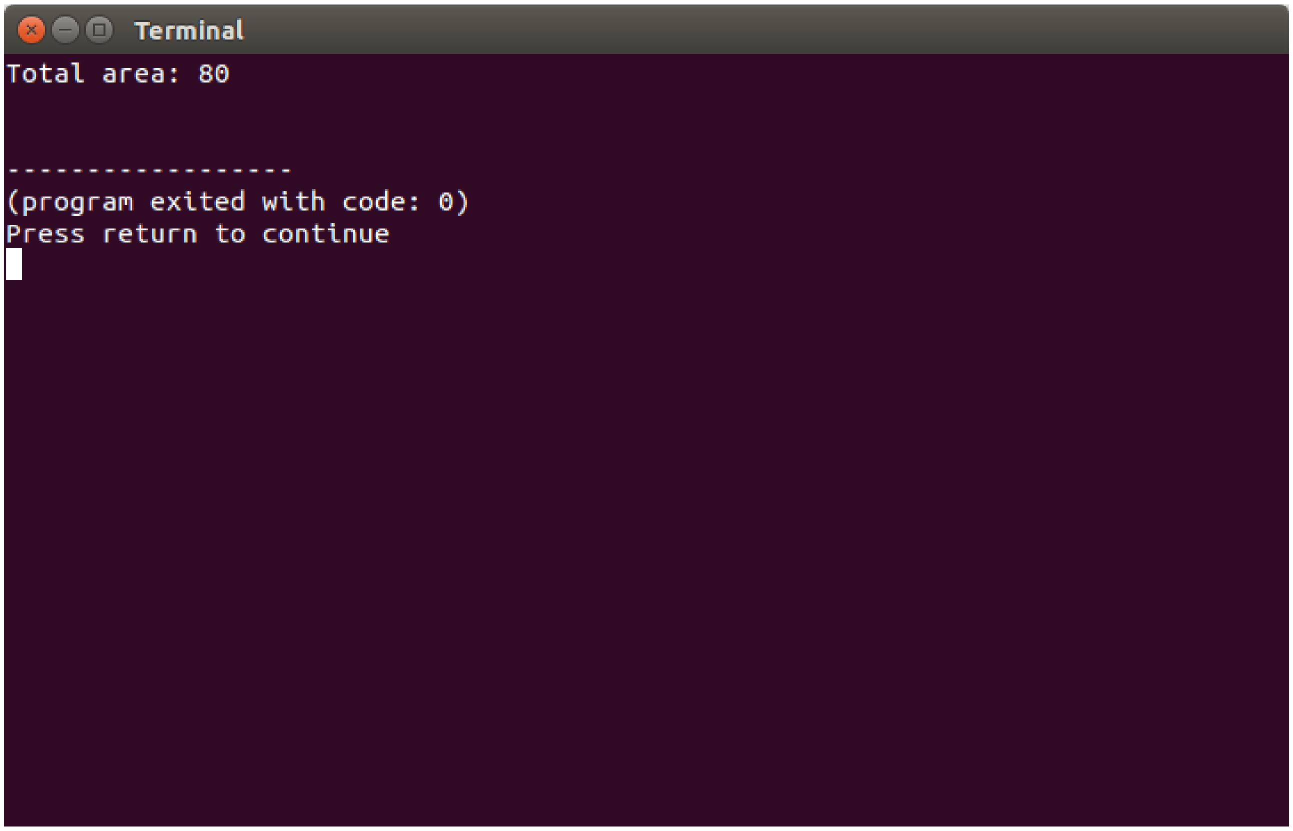
```
#include <iostream>

using namespace std;

// Base class
class Shape
{
   public:
      void setWidth(int w)
      {
         width = w;
      }
      void setHeight(int h)
      {
         height = h;
      }
   protected:
      int width;
      int height;
};

// Derived class
class Rectangle: public Shape
{
   public:
      int getArea()
      {
         return (width * height);
      }
};

int main(void)
{
   Rectangle Rect;

   Rect.setWidth(8);
   Rect.setHeight(10);

   // Print the area of the object.
   cout << "Total area: " << Rect.getArea() << endl;

   return 0;
}
```

PROCEDURE:

1.Open CodeBlocks

2.Create new project and open console application
3.Enter source code
4.Run and build Project
5.The output is obtained

OUTPUT:



```
Total area: 80


------------------
(program exited with code: 0)
Press return to continue
```

RESULT: Hence, the program to demonstrate single inheritance is implemented.

SOURCE:http://www.tutorialspoint.com/cplusplus/cpp_inheritance.htm

12.<u>MULTIPLE INHERITANCE:</u>

AIM: Program to demonstrate multiple inheritance

APPARATUS:
1.Code Blocks
2.Ubuntu OS

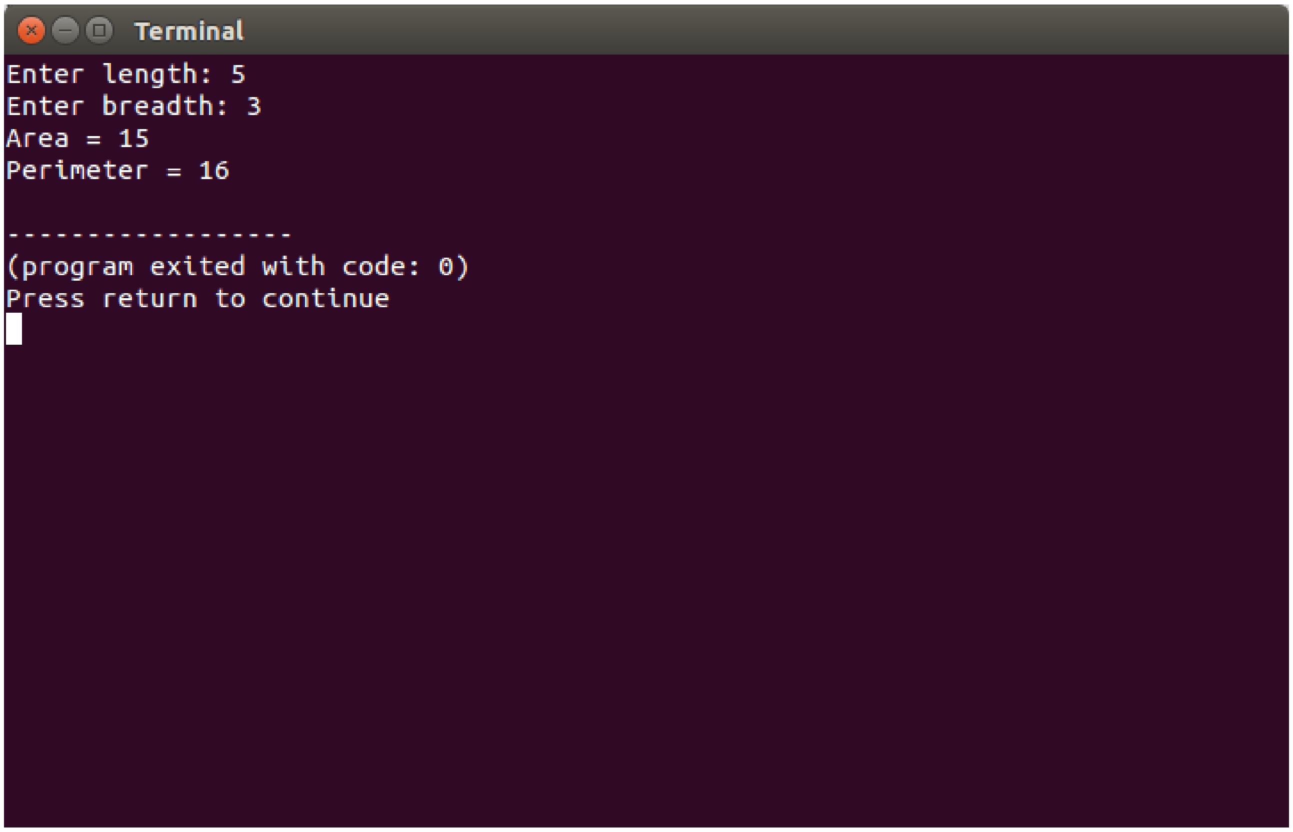
SOURCE CODE:

```
#include <iostream>
using namespace std;
class Area
{
  public:
    float area_calc(float l,float b)
    {
        return l*b;
    }
};

class Perimeter
{
  public:
    float peri_calc(float l,float b)
    {
        return 2*(l+b);
    }
};

/* Rectangle class is derived from classes Area and Perimeter. */
class Rectangle : private Area, private Perimeter
```

```
{
    private:
        float length, breadth;
    public:
       Rectangle() : length(0.0), breadth(0.0) { }
       void get_data( )
       {
           cout<<"Enter length: ";
           cin>>length;
           cout<<"Enter breadth: ";
           cin>>breadth;
       }

       float area_calc()
       {
       /* Calls area_calc() of class Area and returns it. */

           return Area::area_calc(length,breadth);
       }

       float peri_calc()
       {
       /* Calls peri_calc() function of class Perimeter and returns it.
*/

           return Perimeter::peri_calc(length,breadth);
       }
};

int main()
{
    Rectangle r;
    r.get_data();
    cout<<"Area = "<<r.area_calc();
    cout<<"\nPerimeter = "<<r.peri_calc();
    return 0;
}
```

PROCEDURE:
1.Open CodeBlocks
2.Create new project and open console application
3.Enter source code
4.Run and build Project
5.The output is obtained

OUTPUT:



```
Enter length: 5
Enter breadth: 3
Area = 15
Perimeter = 16

------------------
(program exited with code: 0)
Press return to continue
```

RESULT: Hence, the program to demonstrate multiple inheritance is implemented.

SOURCE:http://www.programiz.com/cpp-programming/multilevel-multiple-inheritance

13.<u>MULTI LEVEL INHERITANCE</u>:

AIM: Program to demonstrate multi level inheritance

APPARATUS:
1.Code Blocks
2.Ubuntu OS

SOURCE CODE:

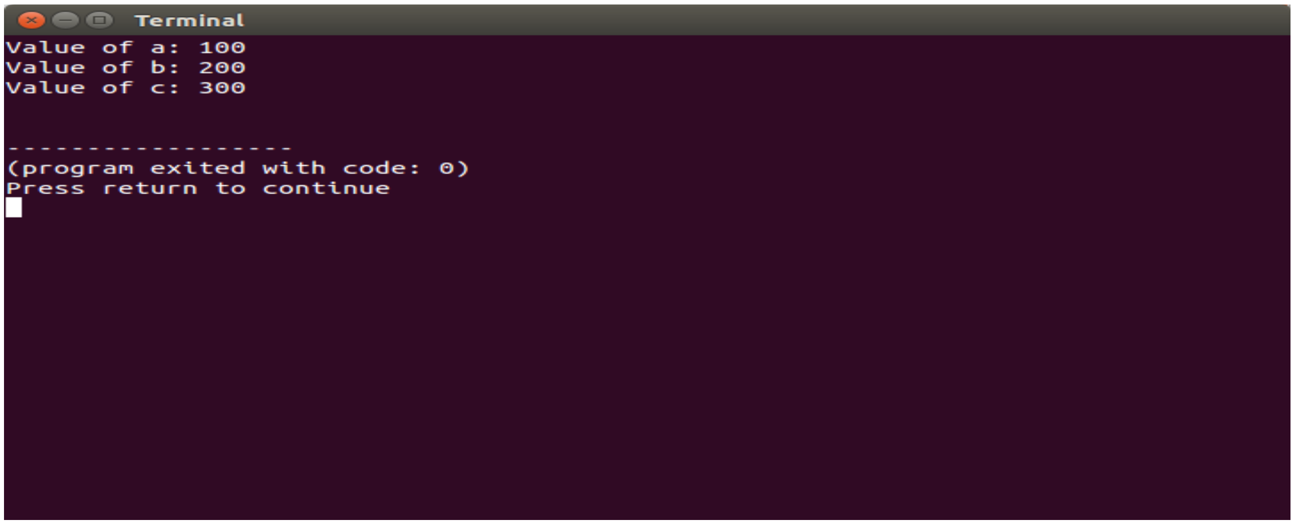
```
#include <iostream>
using namespace std;

//Base Class : class A
class A
{
    private:
        int a;
    public:
        void get_a(int val_a)
        {
            a=val_a;
        }

        void disp_a(void)
        {
            cout << "Value of a: " << a << endl;
        }
```

```
};

//Here Class B is base class for class C
//and Derived class for class A
class B: public A
{
    private:
        int b;
    public:
        //assign value of a from here
        void get_b(int val_a, int val_b)
        {
            //assign value of a by calling function of class A
            get_a(val_a);
            b=val_b;
        }

        void disp_b(void)
        {
            //display value of a
            disp_a();
            cout << "Value of b: " << b << endl;
        }
};

//Here class C is derived class and B is Base class
class C: public B
{
    private:
        int c;
    public:
        //assign value of a from here
        void get_c(int val_a, int val_b,int val_c)
        {
            /*** Multilevel Inheritance ***/
            //assign value of a, bby calling function of class B and
Class A
            //here Class A is inherited on Class B, and Class B in
inherited on Class B
```

```
            get_b(val_a,val_b);
            c=val_c;
        }

        void disp_c(void)
        {
            //display value of a and b using disp_b()
            disp_b();
            cout << "Value of c: " << c << endl;
        }
};

int main()
{
    //create object of final class, which is Class C
    C objC;

    objC.get_c(100,200,300);
    objC.disp_c();
    return 0;
}
```

PROCEDURE:
1.Open CodeBlocks
2.Create new project and open console application
3.Enter source code
4.Run and build Project
5.The output is obtained

OUTPUT:



```
Value of a: 100
Value of b: 200
Value of c: 300


------------------
(program exited with code: 0)
Press return to continue
```

RESULT: Hence, the program to demonstrate multi level inheritance is implemented.

SOURCE:http://www.includehelp.com/cpp-programs/cpp-inheritance-program-to-demonstrate-example-of-multilevel-inheritance.aspx

14.<u>HIERARCHICAL INHERITANCE:</u>

AIM: Program to demonstrate hierarchical inheritance

APPARATUS:
1.Code Blocks
2.Ubuntu OS

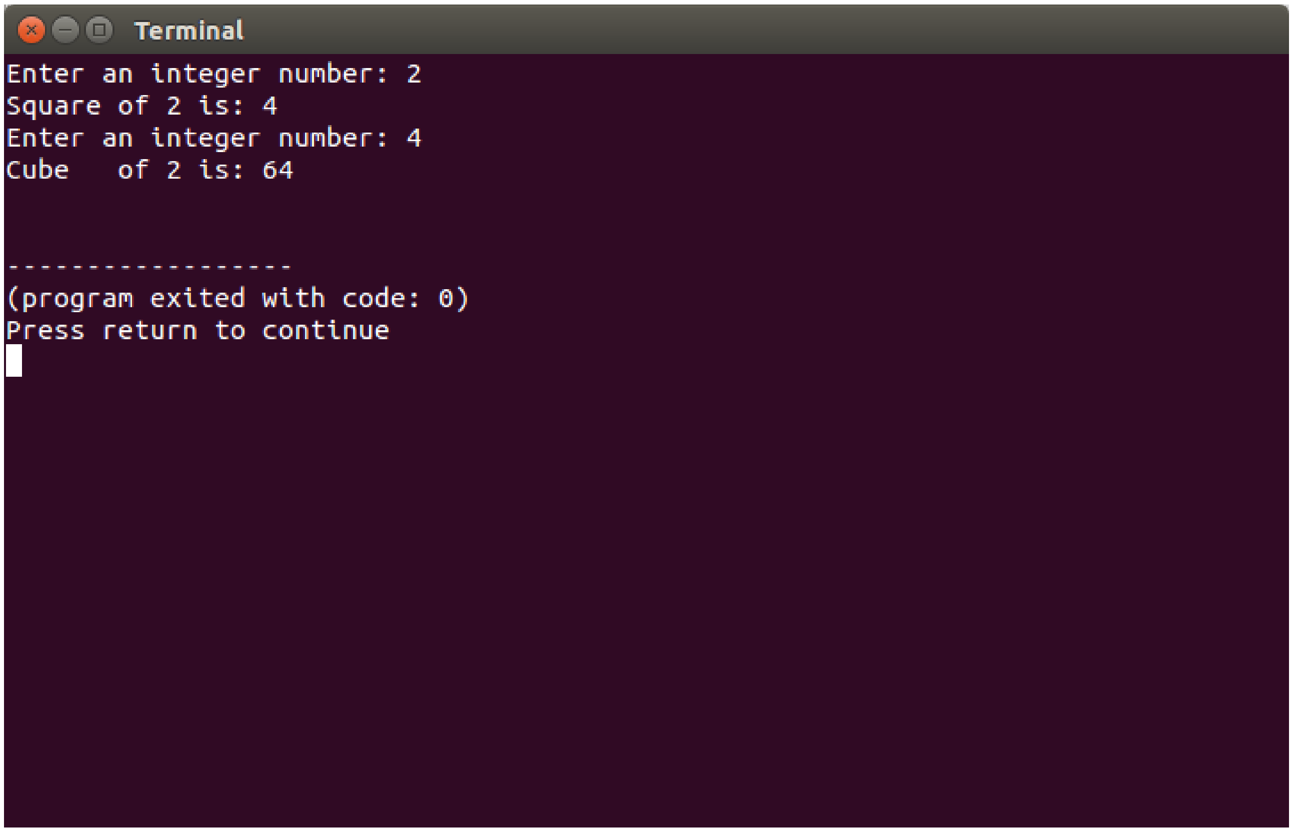
SOURCE CODE:

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

class Number
{
    private:
        int num;
    public:
        void getNumber(void)
        {
            cout << "Enter an integer number: ";
            cin  >> num;
        }
        //to return num
        int returnNumber(void)
        { return num; }
};

//Base Class 1, to calculate square of a number
class Square:public Number
{
    public:
    int getSquare(void)
    {
        int num,sqr;
        num=returnNumber(); //get number from class Number
        sqr=num*num;
        return sqr;
    }
};

//Base Class 2, to calculate cube of a number
class Cube:public Number
{
    private:

    public:
    int getCube(void)
    {
        int num,cube;
        num=returnNumber(); //get number from class Number
        cube=num*num*num;
        return cube;
    }
};
int main()
```

```
{
	Square objS;
	Cube objC;
	int sqr,cube;

	objS.getNumber();
	sqr =objS.getSquare();
	cout << "Square of "<< objS.returnNumber() << " is: " <<
sqr  << endl;

	objC.getNumber();
	cube=objC.getCube();
	cout << "Cube   of "<< objS.returnNumber() << " is: " <<
cube << endl;

	return 0;
}
```

PROCEDURE:
1.Open CodeBlocks
2.Create new project and open console application
3.Enter source code
4.Run and build Project
5.The output is obtained

OUTPUT:



```
Enter an integer number: 2
Square of 2 is: 4
Enter an integer number: 4
Cube   of 2 is: 64


------------------
(program exited with code: 0)
Press return to continue
```

RESULT: Hence, the program to demonstrate hierarchical inheritance is implemented.

SOURCE:http://www.includehelp.com/cpp-programs/cpp-inheritance-program-to-demonstrate-example-of-hierarchical-square-and-cube-of-a-number-inheritance.aspx

15.VIRTUAL BASE CLASSES:

AIM: Program to demonstrate virtual base classes

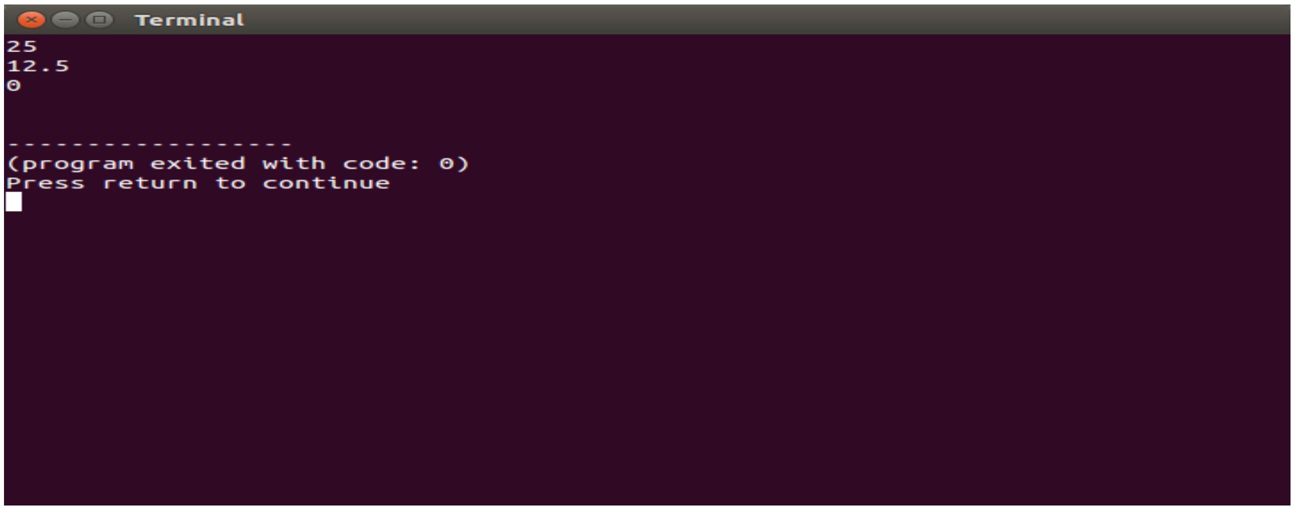
APPARATUS:
1.Code Blocks
2.Ubuntu OS

SOURCE CODE:

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

class Polygon {
  protected:
    float width, height;
  public:
    void set_values (float a, float b)
      { width=a; height=b; }
    virtual float area ()
      { return 0; }
};

class Rectangle: public Polygon {
  public:
    float area ()
      { return width * height; }
};

class Triangle: public Polygon {
  public:
    float area ()
      { return (width * height / 2); }
};

int main () {
  Rectangle rect;
  Triangle trgl;
  Polygon poly;
  Polygon * ppoly1 = &rect;
  Polygon * ppoly2 = &trgl;
  Polygon * ppoly3 = &poly;
  ppoly1->set_values (5,5);
  ppoly2->set_values (5,5);
  ppoly3->set_values (5,5);
  cout << ppoly1->area() << '\n';
  cout << ppoly2->area() << '\n';
  cout << ppoly3->area() << '\n';
  return 0;
}
```

PROCEDURE:
1.Open CodeBlocks
2.Create new project and open console application
3.Enter source code
4.Run and build Project
5.The output is obtained

OUTPUT:



```
25
12.5
0


------------------
(program exited with code: 0)
Press return to continue
```

RESULT: Hence, the program to demonstrate virtual base classes is implemented.

SOURCE:http://www.cplusplus.com/doc/tutorial/polymorphism/

16.ABSTRACT CLASS:

AIM: Program to demonstrate abstract classes

APPARATUS:
1.Code Blocks
2.Ubuntu OS

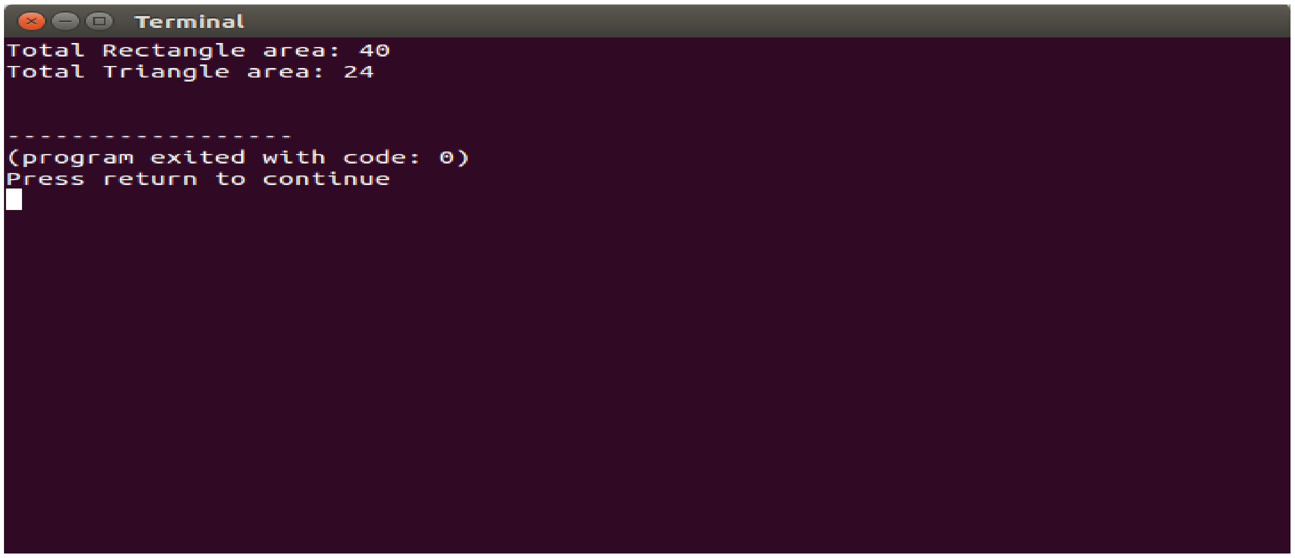
SOURCE CODE:

```
#include <iostream>

using namespace std;

// Base class
class Shape
{
public:
   // pure virtual function providing interface framework.
   virtual double getArea() = 0;
   void setWidth(double w)
   {
      width = w;
   }
   void setHeight(double h)
   {
      height = h;
   }
protected:
   double width;
  double height;
};

// Derived classes
class Rectangle: public Shape
{
public:
   double getArea()
   {
      return (width * height);
   }
};
class Triangle: public Shape
{
public:
   double getArea()
   {
      return (width * height)/2;
   }
};

int main(void)
{
   Rectangle Rect;
   Triangle  Tri;

   Rect.setWidth(8);
   Rect.setHeight(5);
   // Print the area of the object.
   cout << "Total Rectangle area: " << Rect.getArea() << endl;

   Tri.setWidth(4);
   Tri.setHeight(12);
   // Print the area of the object.
   cout << "Total Triangle area: " << Tri.getArea() << endl;

   return 0;
}
```

PROCEDURE:

1.Open CodeBlocks
2.Create new project and open console application
3.Enter source code
4.Run and build Project
5.The output is obtained

OUTPUT:



```
Total Rectangle area: 40
Total Triangle area: 24


------------------
(program exited with code: 0)
Press return to continue
```

RESULT: Hence, the program to demonstrate virtual base classes is implemented.

SOURCE:http://www.tutorialspoint.com/cplusplus/cpp_interfaces.htm

17.ARRAY OF POINTERS:

AIM: Program to demonstrate array of pointers.

APPARATUS:
1. Code Blocks
2. Ubuntu OS

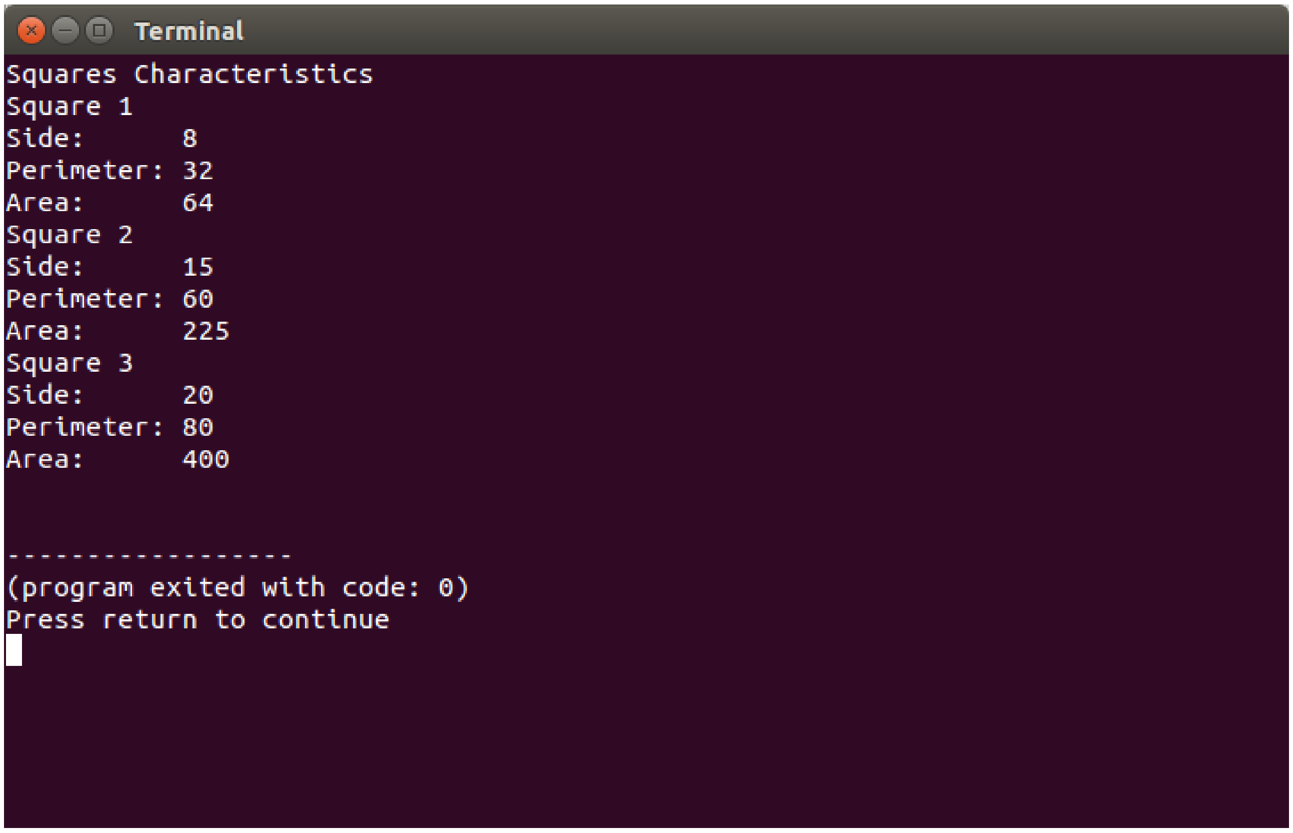
SOURCE CODE :

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

class CSquare
{
public:
    double Side;

    CSquare() : Side(0.00) {}
    CSquare(double side) : Side(side) { }
    ~CSquare() { }

    double getSide() const { return Side; }
    void setSide(const double s)
    {
        if( s <= 0 )
                Side = 0.00;
        else
                Side = s;
        }

    double Perimeter() { return Side * 4; }
    double Area() { return Side * Side; }
};

int main()
{
    CSquare *sqr[3];
    sqr[0] = new CSquare;
    sqr[0]->setSide(8);
    sqr[1] = new CSquare;
    sqr[1]->setSide(15);
    sqr[2] = new CSquare;
    sqr[2]->setSide(20);

    cout << "Squares Characteristics" << endl;
    cout << "Square 1" << endl;
    cout << "Side:      " << sqr[0]->getSide() << endl;
    cout << "Perimeter: " << sqr[0]->Perimeter() << endl;
    cout << "Area:      " << sqr[0]->Area() << endl;

    cout << "Square 2" << endl;
    cout << "Side:      " << sqr[1]->getSide() << endl;
    cout << "Perimeter: " << sqr[1]->Perimeter() << endl;
    cout << "Area:      " << sqr[1]->Area() << endl;

    cout << "Square 3" << endl;
    cout << "Side:      " << sqr[2]->getSide() << endl;
    cout << "Perimeter: " << sqr[2]->Perimeter() << endl;
    cout << "Area:      " << sqr[2]->Area() << endl;


    return 0;
}
```

PROCEDURE:

1. Open code blocks IDE .
2. Create a new project and open console application.
3. Enter the source code
4. Check for errors.
5. Run and build the project.
6. Hence,ouput is obtained.

OUTPUT:



```
Squares Characteristics
Square 1
Side:      8
Perimeter: 32
Area:      64
Square 2
Side:      15
Perimeter: 60
Area:      225
Square 3
Side:      20
Perimeter: 80
Area:      400


------------------
(program exited with code: 0)
Press return to continue
```

RESULT: Hence, program to demostrate array of pointers is executed using.

SOURCE:http://www.functionx.com/cpp/examples/arrayofpointers1.htm

18.<u>SUB STRING IN THE GIVEN STRING</u>:

AIM: Program to find the number of times a sub string is available in the given string

APPARATUS:

1. Code Blocks
2. Ubuntu OS

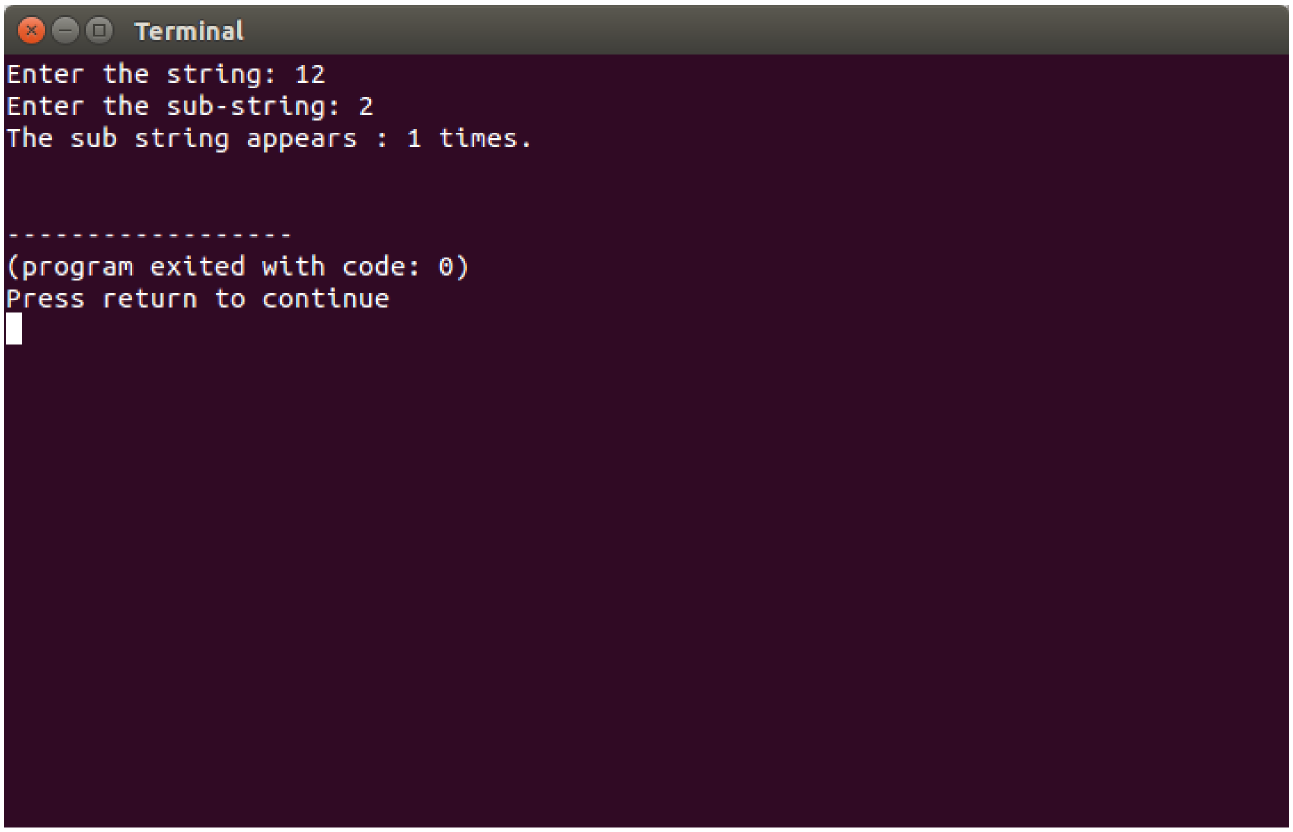
SOURCE CODE :

```
#include <iostream>
#include <string>
using namespace std;
// returns count of non-overlapping occurrences of 'sub' in 'str'
int countSubstring(const string& str, const string& sub)
{
    if (sub.length() == 0) return 0;
    int count = 0;
    for (size_t offset = str.find(sub);offset != string::npos; offset =
str.find(sub, offset + sub.length()))
    {
        ++count;
    }
    return count;
}

int main()
{
        string str,sub;
        cout<< "Enter the string: ";
        getline(cin,str);

        cout<< "Enter the sub-string: ";
        cin >> sub;
    cout << "The sub string appears : "<< countSubstring(str ,
sub) << " times." << '\n';
```

```
    return 0;
}
```

PROCEDURE:
1. Open code blocks IDE .
2. Create a new project and open console application.
3. Enter the source code
4. Check for errors.
5. Run and build the project.
6. Hence,ouput is obtained.

OUTPUT:



```
Enter the string: 12
Enter the sub-string: 2
The sub string appears : 1 times.


------------------
(program exited with code: 0)
Press return to continue
```

RESULT: Hence, program to find the number of times a sub string is available in the given string is executed.

SOURCE:https://tfetimes.com/c-count-occurrences-of-a-substring/

19.<u>POINTERS TO FUNCTIONS & TO OBJECTS</u>:

AIM: Program to demonstrate pointers to functions and objects

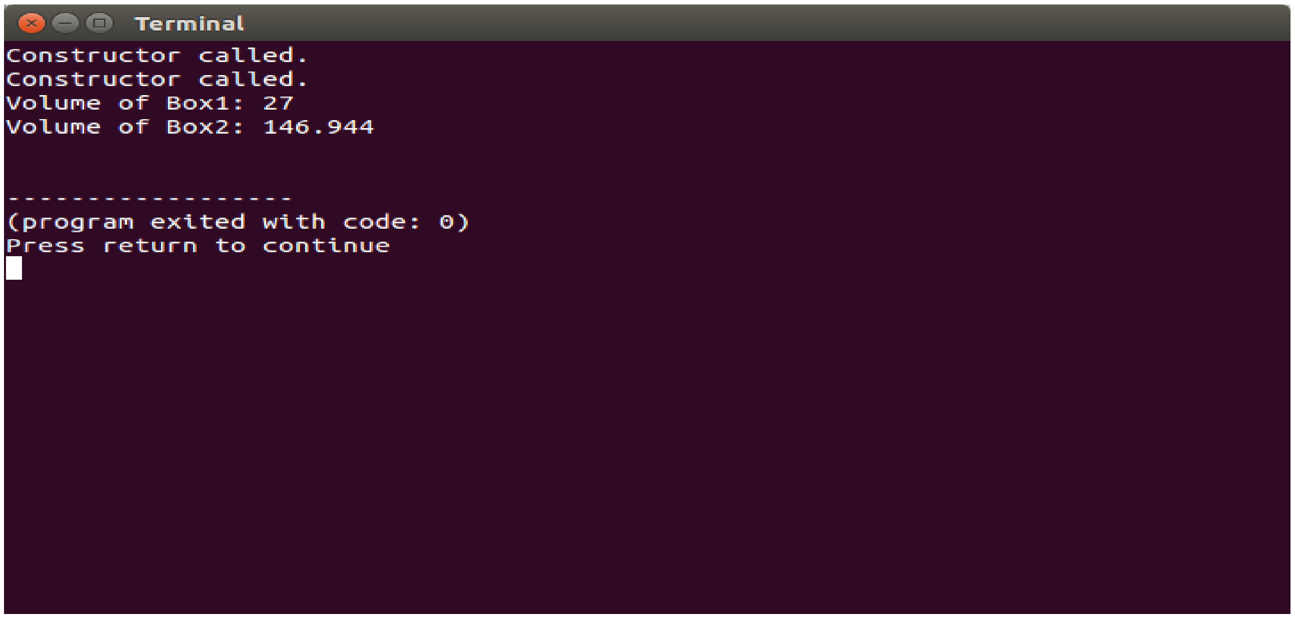
APPARATUS:
1. Code Blocks
2. Ubuntu OS

SOURCE CODE :
```
#include <iostream>

using namespace std;

class Box
{
   public:
      // Constructor definition
      Box(double l=2.0, double b=2.0, double h=2.0)
      {
         cout <<"Constructor called." << endl;
         length = l;
         breadth = b;
         height = h;
      }
      double Volume()
      {
         return length * breadth * height;
      }
   private:
      double length;     // Length of a box
      double breadth;    // Breadth of a box
      double height;     // Height of a box
};

int main(void)
```

```
{
   Box Box1(3, 3, 3);    // Declare box1
   Box Box2(8.2, 6.4, 2.8);    // Declare box2
   Box *ptrBox;                // Declare pointer to a class.

   // Save the address of first object
   ptrBox = &Box1;

   // Now try to access a member using member access operator
   cout << "Volume of Box1: " << ptrBox->Volume() << endl;

   // Save the address of first object
   ptrBox = &Box2;

   // Now try to access a member using member access operator
   cout << "Volume of Box2: " << ptrBox->Volume() << endl;

   return 0;
}
```

PROCEDURE:
1. Open code blocks IDE .
2. Create a new project and open console application.
3. Enter the source code
4. Check for errors.
5. Run and build the project.
6. Hence,ouput is obtained.

OUTPUT:



```
Constructor called.
Constructor called.
Volume of Box1: 27
Volume of Box2: 146.944


------------------
(program exited with code: 0)
Press return to continue
```

RESULT: Hence, program to demonstrate pointers to functions and objects is executed.

SOURCE: http://www.tutorialspoint.com/cplusplus/cpp_pointer_to_class.htm

20.<u>EXCEPTION HANDLING CONCEPT</u>:

AIM : To Implement exception handling concept using a division by zero program.

APPARATUS:

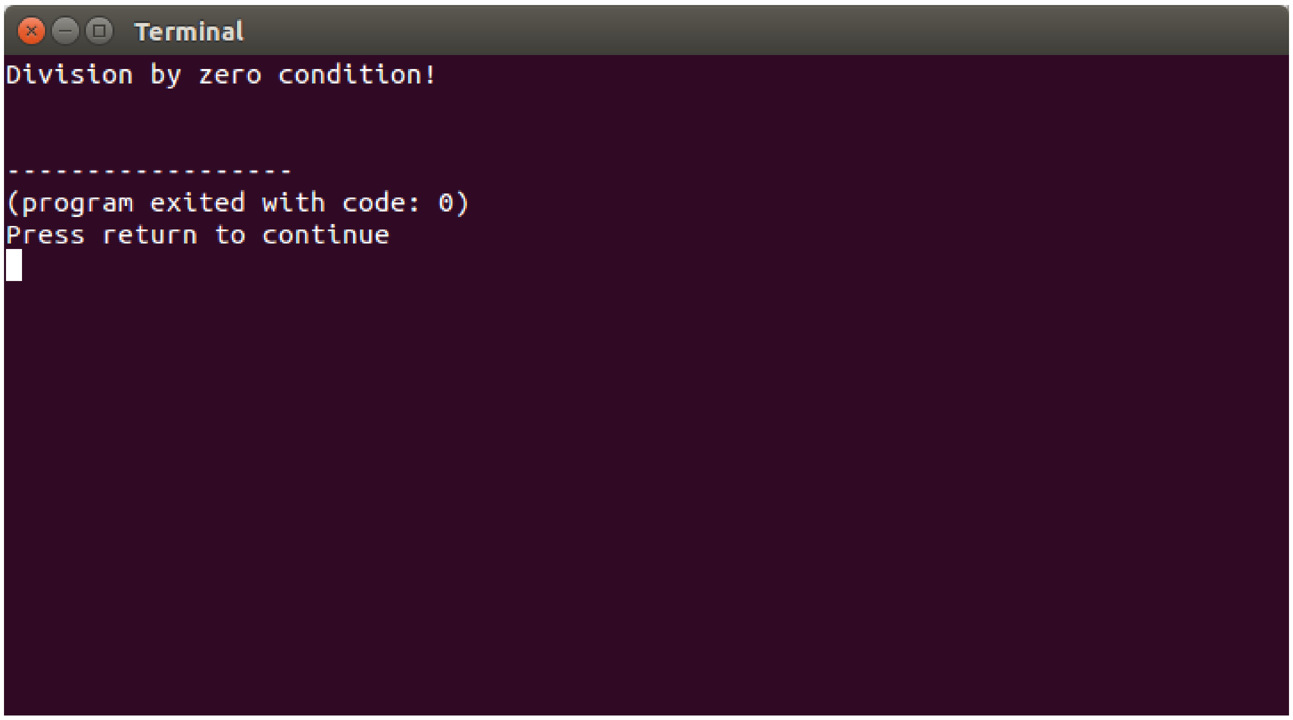
1.Debian OperatingSystem
2.CodeBlocks IDE software

SOURCE CODE :

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

double division(int a, int b)
{
   if( b == 0 )
   {
      throw "Division by zero condition!";
   }
   return (a/b);
}

int main ()
{
   int x = 50;
   int y = 0;
   double z = 0;

   try {
     z = division(x, y);
     cout << z << endl;
   }catch (const char* msg) {
     cerr << msg << endl;
   }
   return 0;
}
```

PROCEDURE :
1. Open code blocks IDE .
2. Create a new project and open console application.
3. Enter the source code
4. Check for errors.
5. Run and build the project.
6. Hence,ouput is obtained.

OUTPUTS :



```
Division by zero condition!


------------------
(program exited with code: 0)
Press return to continue
```

RESULT :Hence, exception handling is executed for a division by zero case.

SOURCES:http://www.tutorialspoint.com/cplusplus/cpp_exceptions_handling.htm