**AMAN AGRAWAL**

**2020BTECS00074**

**Programming Laboratory-I**

**Assignment No-5**

**(Virtual functions and namespace)**

1. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get\_data() to initialize base class data members and another member function display\_area() to compute and display the area of figures. Make display\_area() as a virtual function and redefine this function in the derived classes to suit their requirements.

Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area. Remember the two values given as input will be treated as lengths of two sides in the case of rectangles, and as base and height in the case of triangles, and used as follows: Area of rectangle = x \* y

Area of triangle = 1/2 \* x \* y

#include<iostream>

using namespace std;

class Shape{

    public:

    double x,y;

    virtual void get\_data(){

        cout<<"Enter the base length: ";

        cin>>x;

        cout<<"Enter the height: ";

        cin>>y;

    }

    virtual void display\_area(){

        cout<<"Area of the shape: "<<x\*y<<endl;

    }

};

class Triangle:public Shape{

    public:

    void get\_data(){

        cout<<endl<<"Triangle:"<<endl;

        cout<<"Enter the base length:";

        cin>>x;

        cout<<"Enter the Height:";

        cin>>y;

    }

    void display\_area(){

        cout<<"Area of the Triangle = "<<(x\*y)/2<<endl;

    }

};

class Rectangle:public Shape{

    public:

    void get\_data(){

        cout<<endl<<"Rectangle:"<<endl;

        cout<<"Enter the length:";

        cin>>x;

        cout<<"Enter the breadth:";

        cin>>y;

    }

    void display\_area(){

        cout<<"Area of the Rectangle = "<<x\*y<<endl;

    }

};

int main(){

    Triangle t;

    Rectangle r;

    t.get\_data();

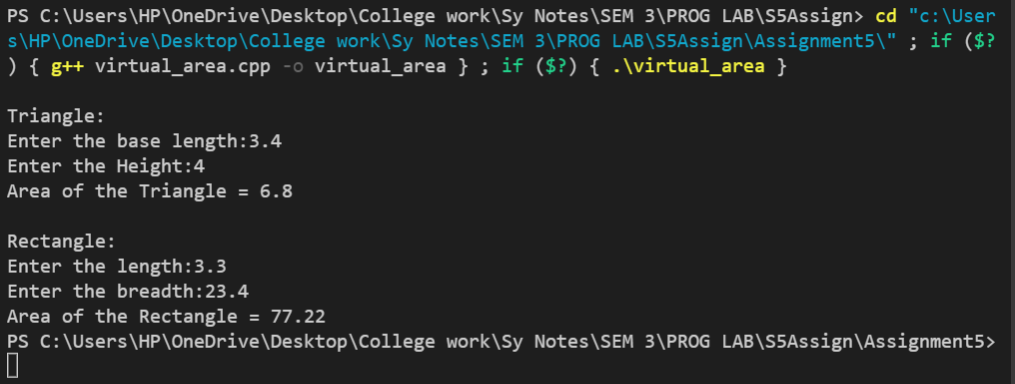
    t.display\_area();

    r.get\_data();

    r.display\_area();

    return 0;

}

Output: 

1. Extend the above program to display the area of circles. This requires addition of a new derived class 'circle' that computes the area of a circle. Remember, for a circle we need only one value, its radius, but the get\_data() function in the base class requires two values to be passed. (Hint: Make the second argument of get\_data() function as a default one with zero value.)

#include<iostream>

using namespace std;

class Shape{

    public:

    double x,y;

    virtual void get\_data(){

        cout<<"Enter the base length: ";

        cin>>x;

        cout<<"Enter the height: ";

        cin>>y;

    }

    virtual void display\_area(){

        cout<<"Area of the shape: "<<x\*y<<endl;

    }

};

class Triangle:public Shape{

    public:

    void get\_data(){

        cout<<endl<<"Triangle:"<<endl;

        cout<<"Enter the base length:";

        cin>>x;

        cout<<"Enter the Height:";

        cin>>y;

    }

    void display\_area(){

        cout<<"Area of the Triangle = "<<(x\*y)/2<<endl;

    }

};

class Rectangle:public Shape{

    public:

    void get\_data(){

        cout<<endl<<"Rectangle:"<<endl;

        cout<<"Enter the length:";

        cin>>x;

        cout<<"Enter the breadth:";

        cin>>y;

    }

    void display\_area(){

        cout<<"Area of the Rectangle = "<<x\*y<<endl;

    }

};

class Circle:public Shape{

    public:

    void get\_data(){

        cout<<endl<<"Circle:"<<endl;

        cout<<"Enter the radius:";

        cin>>x;

        y=0;

    }

    void display\_area(){

        cout<<"Area of the Rectangle = "<<3.14\*x\*x<<endl;

    }

};

int main(){

    Triangle t;

    Rectangle r;

    Circle c;

    t.get\_data();

    t.display\_area();

    r.get\_data();

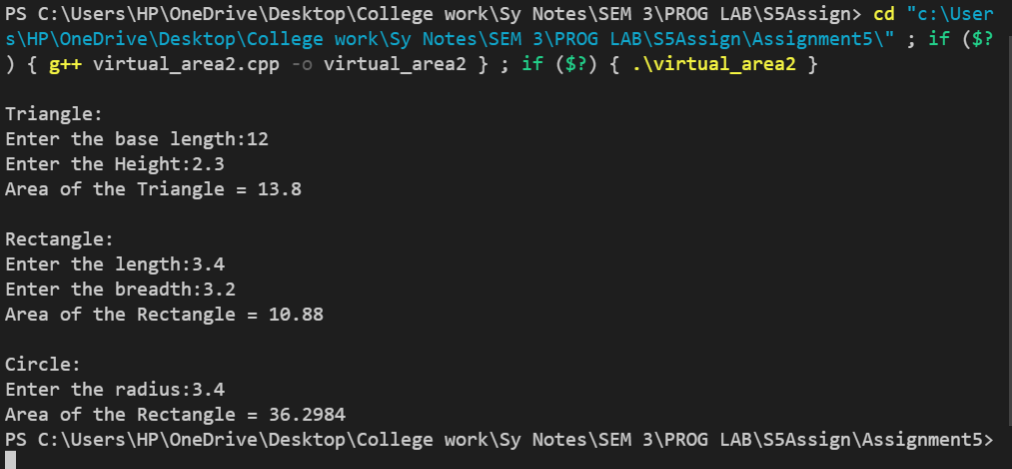
    r.display\_area();

    c.get\_data();

    c.display\_area();

    return 0;

}

Output: 

1. Run the above program with the following modifications:
2. Remove the definition of display area() from one of the derived classes.
3. In addition to the above change, declare the display\_area() as virtual in the base class shape.

#include<iostream>

using namespace std;

class Shape{

    public:

    double x,y;

    virtual void get\_data(){

        cout<<"Enter the base length: ";

        cin>>x;

        cout<<"Enter the height: ";

        cin>>y;

    }

    virtual void display\_area(){

        cout<<"Area of the shape: "<<x\*y<<endl;

    }

};

class Triangle:public Shape{

    public:

    void get\_data(){

        cout<<endl<<"Triangle:"<<endl;

        cout<<"Enter the base length:";

        cin>>x;

        cout<<"Enter the Height:";

        cin>>y;

    }

    void display\_area(){

        cout<<"Area of the Triangle = "<<(x\*y)/2<<endl;

    }

};

class Rectangle:public Shape{

    public:

    void get\_data(){

        cout<<endl<<"Rectangle:"<<endl;

        cout<<"Enter the length:";

        cin>>x;

        cout<<"Enter the breadth:";

        cin>>y;

    }

};

class Circle:public Shape{

    public:

    void get\_data(){

        cout<<endl<<"Circle:"<<endl;

        cout<<"Enter the radius:";

        cin>>x;

        y=0;

    }

    void display\_area(){

        cout<<"Area of the Rectangle = "<<3.14\*x\*x<<endl;

    }

};

int main(){

    Triangle t;

    Rectangle r;

    Circle c;

    t.get\_data();

    t.display\_area();

    r.get\_data();

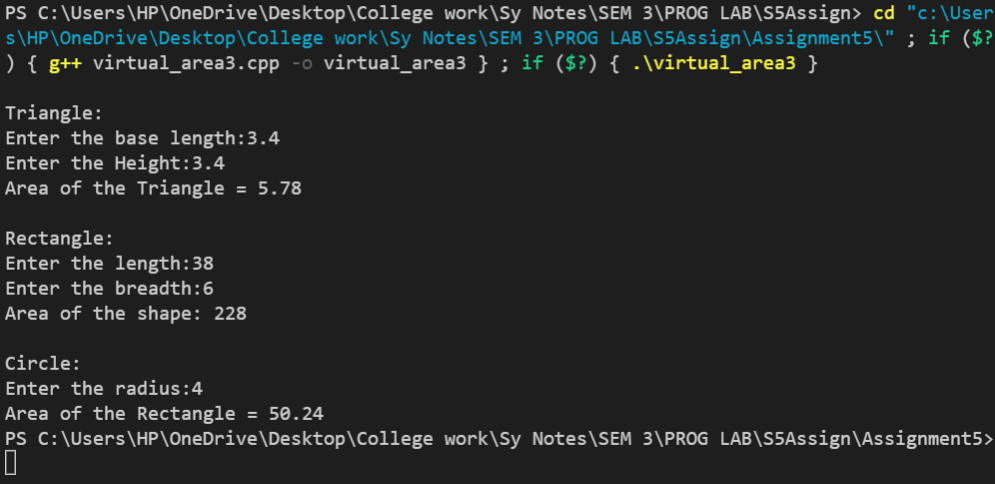
    r.display\_area();

    c.get\_data();

    c.display\_area();

    return 0;

}

Output: 

1. Create two namespaces as NMS1 and NMS2, define variable ‘a’ and function seta() and geta () in NMS1 and define variable ‘b’ and function setb() and getb () in NMS2. Write a program to
2. Assign and display data **without using keyword.**

#include<iostream>

using namespace std;

namespace NMS1{

    int a;

    void seta(){

        cout<<endl<<"Enter the value of a: ";

        cin>>a;

    }

    void geta(){

        cout<<endl<<"a = "<<a<<endl;

    }

}

namespace NMS2{

    int b;

    void setb(){

        cout<<endl<<"Enter the value of b: ";

        cin>>b;

    }

    void getb(){

        cout<<endl<<"b= "<<b<<endl;

    }

}

int main(){

    //without using keyword

    NMS1::seta();

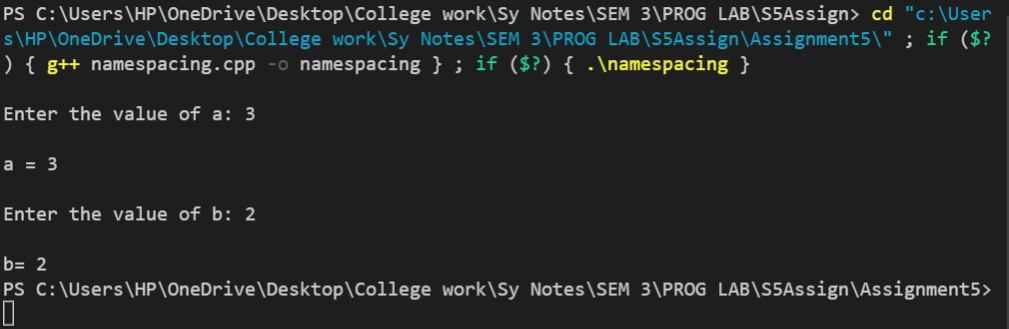
    NMS1::geta();

    NMS2::setb();

    NMS2::getb();

return 0;

}

Output: 

1. Assign and display data **with using keyword.**

#include<iostream>

using namespace std;

namespace NMS1{

    int a;

    void seta(){

        cout<<endl<<"Enter the value of a: ";

        cin>>a;

    }

    void geta(){

        cout<<endl<<"a = "<<a<<endl;

    }

}

namespace NMS2{

    int b;

    void setb(){

        cout<<endl<<"Enter the value of b: ";

        cin>>b;

    }

    void getb(){

        cout<<endl<<"b= "<<b<<endl;

    }

}

int main(){

    //Using keyword

    using namespace NMS1;

    seta();

    geta();

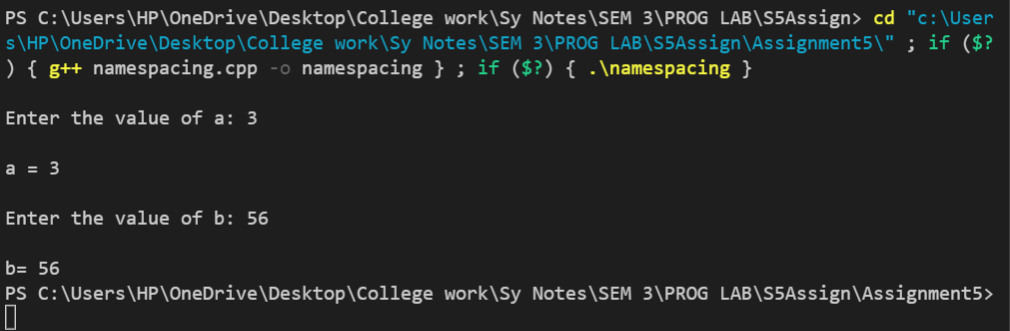
    using namespace NMS2;

    setb();

    getb();

return 0;

}

Output: 

1. Create two namespaces as Outer and Inner, add any one variable, any one function in both the namespaces.

Write a program to display members of both Outer and Inner namespace.

#include<iostream>

using namespace std;

namespace Outer{

    string out = "4:00 pm";

    void outtime(){

        cout<<"Out at "<<out<<endl;

    }

}

namespace Inner{

    string in = "9:00 am";

    void intime(){

        cout<<"In at "<<in<<endl;

    }

}

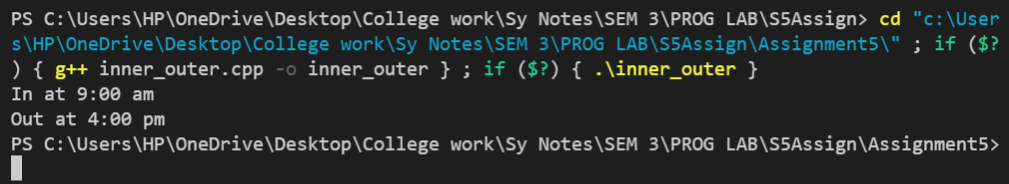
int main(){

    Inner::intime();

    Outer::outtime();

return 0;

}

Output: 

1. Create a class Employee in namespace Organization. Create functions as setDetails (EmpID, Name, Salary) and getDetails ().

Write a program to call both the functions using object of Employee class.

#include<iostream>

using namespace std;

namespace employ{

    class Employee{

    public:

    int EmpId, empSalary;

    string empname;

    void setDetails(){

        cout<<"Enter the details:"<<endl;

        cout<<"Id: ";

        cin>>EmpId;

        cout<<"Name: ";

        cin>>empname;

        cout<<"Salary: ";

        cin>>empSalary;

    }

    void getDetails(){

        cout<<"Employee Details:"<<endl;

        cout<<"Id: "<<EmpId<<endl<<"Name: "<<empname<<endl<<"Salary: "<<empSalary<<endl;

    }

    };

}

int main(){

    using namespace employ;

    Employee emp;

    emp.setDetails();

    emp.getDetails();

return 0;

}

Output: 