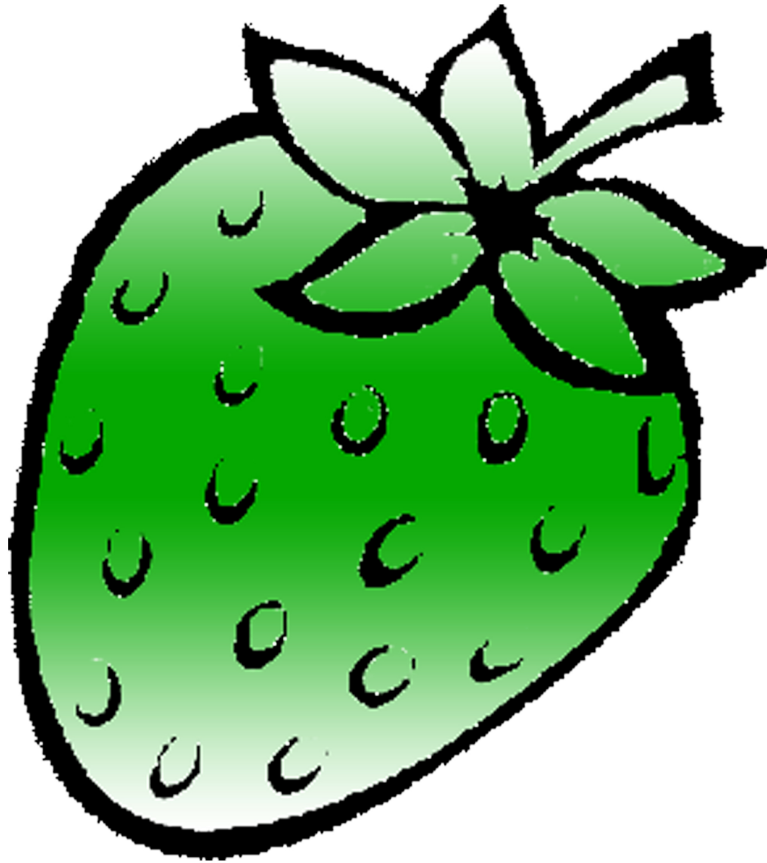


# STRAWBERRY



*/strawberrydevelopers*



*/strawberry\_app*

*For more visit:*

*Strawberrydevelopers.weebly.com*

# Experiment No. 9

## PART A

### (PART A: TO BE REFERRED BY STUDENTS)

#### A.1 Aim: To study virtual functions and Polymorphism

**P1:** Create a base class called 'SHAPE' having

- two data members of type double
- member function *get-data*( ) to initialize base class data members - pure virtual member function *display-area*( ) to compute and display the area of the geometrical object.

Derive two specific classes 'TRIANGLE' and 'RECTANGLE' from the base class. Using these three classes design a program that will accept dimension of a triangle / rectangle interactively and display the area.

**P2:** Extend the above program to display the area of circles. This requires addition of new derived class 'circle' that computes the area of a circle. Remember, for a circle we need only value, its radius, but the *get\_data* function in the base class required two values to be used.

Hint: Make the second argument of *get\_data* function as a default one with zero value.

**P3:** Run the above program with the following modifications:

- (a) Remove the definition of *display\_area*() from one of the derived classes.
- (b) In addition to the above change, declare the *display\_Area*() as virtual in the base class shape.

#### A.2 Prerequisite:

Knowledge of classes and object and concept of inheritance.

#### A.3 Outcome:

After successful completion of this experiment students will be able to

1. Know the polymorphism.
2. Understand the use of virtual and pure virtual functions.

#### A.4 Theory:

##### Virtual Function:

When the function names are same in both the base class and derived class , the function in base class is declared as virtual.

Developed By Strawberry

When a function is made virtual, C++ determines which function to use at run time based on the type of object pointed to by the base pointer , rather than the type of the pointer.

Virtual Function is used to exhibit run time polymorphism in C++ .

### **Pure Virtual Function:**

It is a normal practice to declare a function virtual inside the base class and redefine it in the derived class.

The function inside the base class is seldom used for performing any task. It only serves as a placeholder. Such functions are called “do-nothing” functions

“Do- nothing” functions are called as Pure Virtual Functions

A pure virtual function is defined as follows:

```
virtual void display() = 0;
```

A pure virtual function is a function declared in a base class that has no definition relative to the base class

For pure virtual function , compiler requires each derived class to either define the function or redeclare it as a pure virtual function

A class containing pure virtual functions cannot be used to declare any object of its own. Such class is called as an abstract base class

The main objective of an abstract base class is to provide some traits to the derived classes and to create a base pointer required for achieving run time polymorphism

For Eg:-

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
class A
```

```
{
```

```
    public : int x,y;
```

```
        virtual void getdata (int ,int ){}; //virtual function
```

```
        virtual void display () = 0; //pure virtual
```

```
};
```

**Developed By Strawberry**

```

class sum:public A
{
    public: void getdata (int , int);
           void display ();
};
class mul:public A
{
    public: void getdata (int , int);
           void display ();
};
void sum::getdata (int a, int b)
{
    x= a;
    y = b;
}
void sum::display ()
{
    cout<<"\n Addition = "<<x + y;
}
void mul::getdata (int a, int b)
{
    x= a;
    y = b;
}
void mul::display ()
{
    cout<<"\n Multiplication = "<<x * y;
}

void main()
{
    A *a;
    sum s;
    mul m;
    a=&s;
    a->getdata (10,20);
    a->display();
    a=&m;
    a->getdata (10,20);
    a->display();
}

```

Developed By Strawberry

```
    getch();  
}
```

### **Default arguments:**

A function can be called without specifying all its arguments. This won't work on just any function: The function declaration must provide default values for those arguments that are not specified.

For example:

```
#include<iostream.h>  
void main()  
{  
    void print_line(char='*', int= 10);  
    print_line();  
    print_line('=');  
    print_line('&',5);  
}  
void print_line(char ch, int no)  
{  
    for(int i=0;i<no;i++)  
    {  
        cout<<ch;  
        cout<<endl;  
    }  
}
```

Output:

```
*****  
=====
```

## PART B

### (PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll No. N008	Name: AKSHAY BANDA
Program: MBA TECh	Division: C
Semester: 2	Batch : C1
Date of Experiment: 25/3/15	Date of Submission:
Grade :	

B.1 Software Code written by student:

*(Paste your C++ code completed during the 2 hours of practical in the lab here)*

1.

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

```
class shape
{
protected:
    double a,b;
public:
    void getdata()
    {
        cout<<"\nEnter dimensions of 1st side : ";
        cin>>a;
        cout<<"\nEnter dimensions of 2nd side : ";
        cin>>b;
    }
    virtual void display()=0;
};
```

```
class rectangle: public shape
{
```

Developed By Strawberry

```

    double area;
public:
    void display()
    {
        area=a*b;
        cout<<"\nArea of rectangle is : "<<area;
    }
};

```

```

class triangle: public shape
{
    double area;
public:
    void display()
    {
        area=a*b*0.5;
        cout<<"\nArea of triangle is : "<<area;
    }
};

```

```

int main()
{
    shape *s;
    int c;
    cout<<"\nEnter 1 for Rectangle";
    cout<<"\nEnter 2 for triangle ";
    cout<<"\nEnter your choice : ";
    cin>>c;
    if(c==1)
    {
        rectangle r;
        s=&r;
        s->getdata();
        s->display();
    }
    else if(c==2)
    {
        triangle t;
        t.getdata();
        t.display();
    }
}

```

Developed By Strawberry

```
    }  
    else  
    {  
        cout<<"\nInvalid choice ";  
    }  
    return 0;  
}
```

2.

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

```
class shape  
{  
protected:  
    double a,b;  
public:  
    void getdata(double a1, double b1)  
    {  
        a=a1;  
        b=b1;  
    }  
    virtual void display()=0;  
};
```

```
class rectangle: public shape  
{  
    double area;  
public:  
    void display()  
    {  
        area=a*b;  
        cout<<"\nArea of rectangle is : "<<area;  
    }  
};
```

```
class triangle: public shape  
{  
    double area;  
public:
```

Developed By Strawberry



```

void display()
{
    area=a*b*0.5;
    cout<<"\nArea of triangle is : "<<area;
}
};

```

```

class circle: public shape
{
    double area;
public:
    void display()
    {
        area=3.14*a*a;
        cout<<"\nArea of circle is : "<<area;
    }
};

```

```

int main()
{
    int c;
    double a,b;
    cout<<"\nEnter 1 for Rectangle";
    cout<<"\nEnter 2 for triangle ";
    cout<<"\nEnter 3 for circle ";
    cout<<"\nEnter your choice : ";
    cin>>c;
    if(c==1)
    {
        rectangle r;
        cout<<"\nEnter length : ";
        cin>>a;
        cout<<"\nEnter base : ";
        cin>>b;
        r.getdata(a,b);
        r.display();
    }
    else if(c==2)
    {

```

Developed By Strawberry

```

    triangle t;
    cout<<"\nEnter length : ";
    cin>>a;
    cout<<"\nEnter base : ";
    cin>>b;
    t.getdata(a,b);
    t.display();
}
else if(c==3)
{
    circle cr;
    cout<<"\nEnter radius : ";
    cin>>a;
    cr.getdata(a,0);
    cr.display();
}
else
{
    cout<<"\nInvalid choice ";
}
return 0;
}

```

3.

```

#include<iostream>
using namespace std;

class shape
{
protected:
    double a,b;
public:
    void getdata(double a1, double b1)
    {
        a=a1;
        b=b1;
    }
    virtual void display()
    {

```

Developed By Strawberry

```

        cout<<"\SORRY!";
    }
};

class rectangle: public shape
{
    double area;
public:
    void display()
    {
        area=a*b;
        cout<<"\nArea of rectangle is : "<<area;
    }
};

class triangle: public shape
{
    double area;
public:
    void display()
    {
        area=a*b*0.5;
        cout<<"\nArea of triangle is : "<<area;
    }
};

class circle: public shape
{
    double area;
};

int main()
{
    int c;
    double a,b;
    cout<<"\nEnter 1 for Rectangle";
    cout<<"\nEnter 2 for triangle ";
    cout<<"\nEnter 3 for circle ";
    cout<<"\nEnter your choice : ";
}

```

Developed By Strawberry

```

cin>>c;
if(c==1)
{
    rectangle r;
    cout<<"\nEnter length : ";
    cin>>a;
    cout<<"\nEnter base : ";
    cin>>b;
    r.getdata(a,b);
    r.display();
}
else if(c==2)
{
    triangle t;
    cout<<"\nEnter length : ";
    cin>>a;
    cout<<"\nEnter base : ";
    cin>>b;
    t.getdata(a,b);
    t.display();
}
else if(c==3)
{
    circle cr;
    cout<<"\nEnter radius : ";
    cin>>a;
    cr.getdata(a,0);
    cr.display();
}
else
{
    cout<<"\nInvalid choice ";
}
return 0;
}

```

B.2 Input and Output:

Developed By Strawberry

*(Paste your program input and output in following format. If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)*

1.

Enter 1 for Rectangle

Enter 2 for triangle

Enter your choice : 1

Enter dimensions of 1st side : 7

Enter dimensions of 2nd side : 7

Area of rectangle is : 49

2.

Enter 1 for Rectangle

Enter 2 for triangle

Enter 3 for circle

Enter your choice : 3

Enter radius : 7

Area of circle is : 153.86

3.

Enter 1 for Rectangle

Enter 2 for triangle

Enter 3 for circle

Enter your choice : 3

Enter radius : 5

SORRY!

Developed By Strawberry

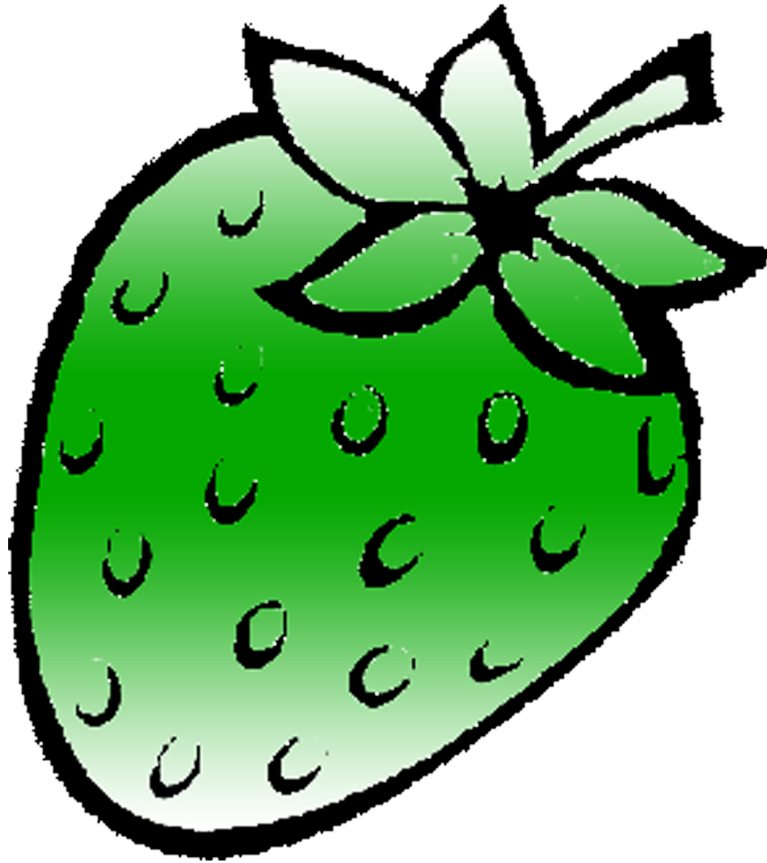
B.3 Conclusion:

*(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.1)*

I learned virtual functions and polymorphism.

Developed By Strawberry

# STRAWBERRY



*/strawberrydevelopers*



*/strawberry\_app*

*For more visit:*

*Strawberrydevelopers.weebly.com*