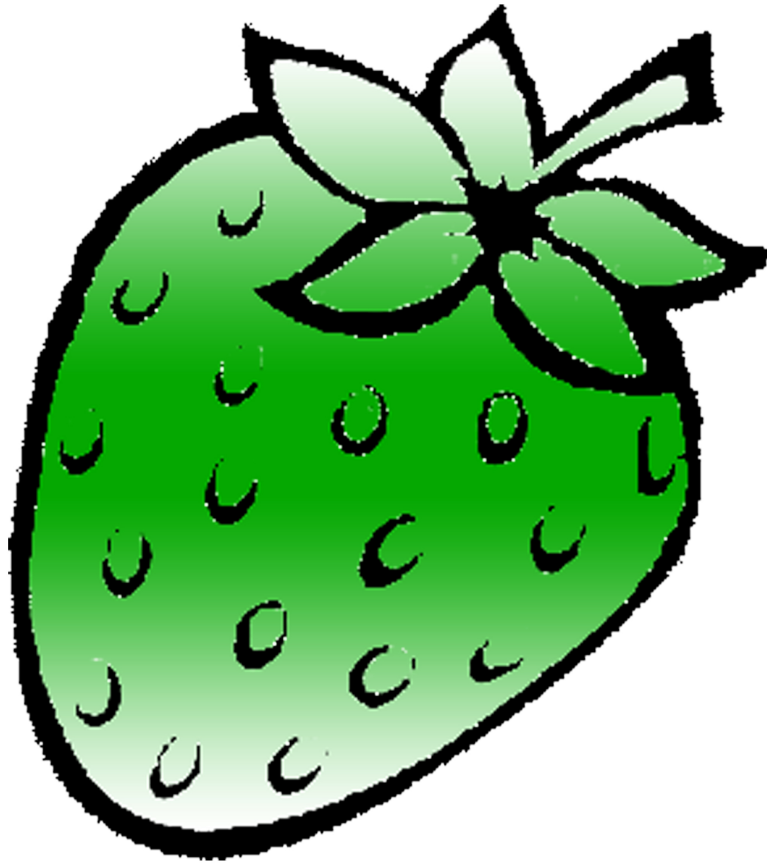


STRAWBERRY



/strawberrydevelopers



/strawberry_app

For more visit:

Strawberrydevelopers.weebly.com

Experiment No. 7

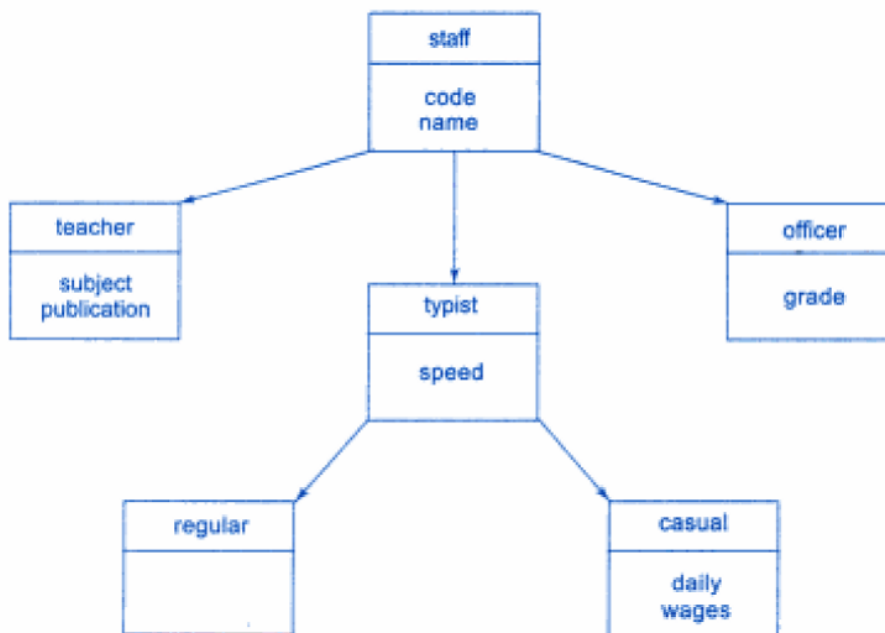
PART A

(PART A: TO BE REFERRED BY STUDENTS)

A.1 Aim: To understand the below concept of Inheritance (Part I)

1. Type of inheritance
2. To understand different access specifier.
3. Constructor in derived classes.

P1: An educational institute wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in figure below. The figure also shows the minimum information required for each class. Specify all the classes and define functions to create the database and retrieve individual information as and when required.



P2: The database created in question 1 does not include educational information of the staff. It has been decided to add this information to teachers and officers (and not for typists) which will help the management in decision making with regard to training, promotion, etc. Add another class called education that holds two pieces of educational information, namely, highest

Developed By Strawberry

educational qualification and highest professional qualification. This class should be inherited by the classes teacher and officer. Modify the program of question 1 to incorporate these additions.

A.2 Prerequisite:

Knowledge of classes and object and concept of inheritance.

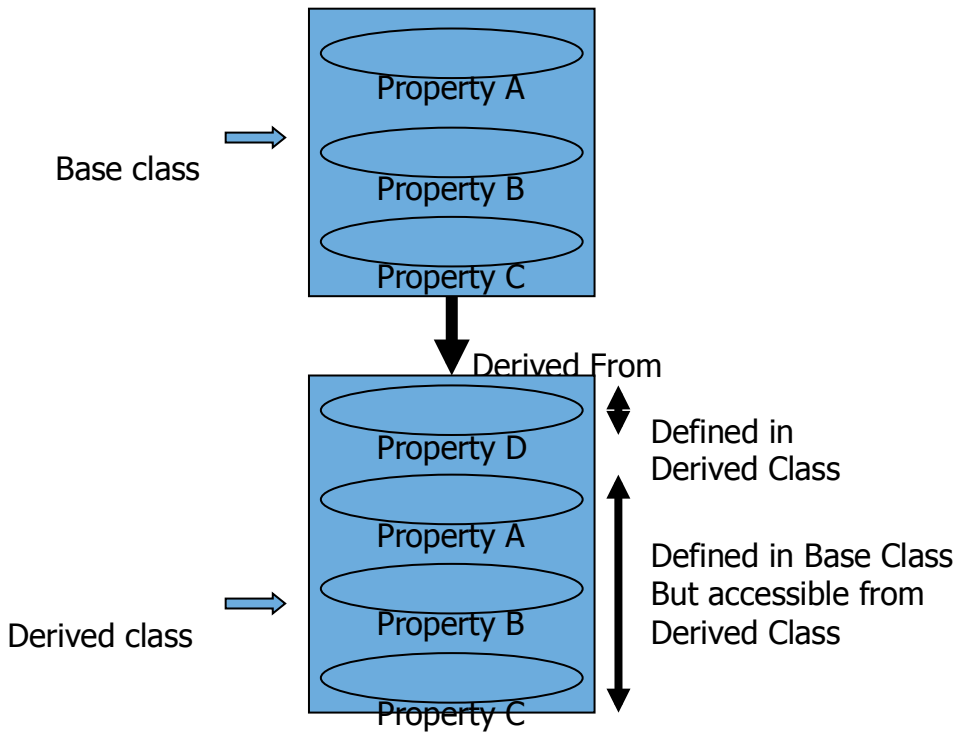
A.3 Outcome:

After successful completion of this experiment students will be able to

1. Know the inheritance implementation

A.4 Theory:

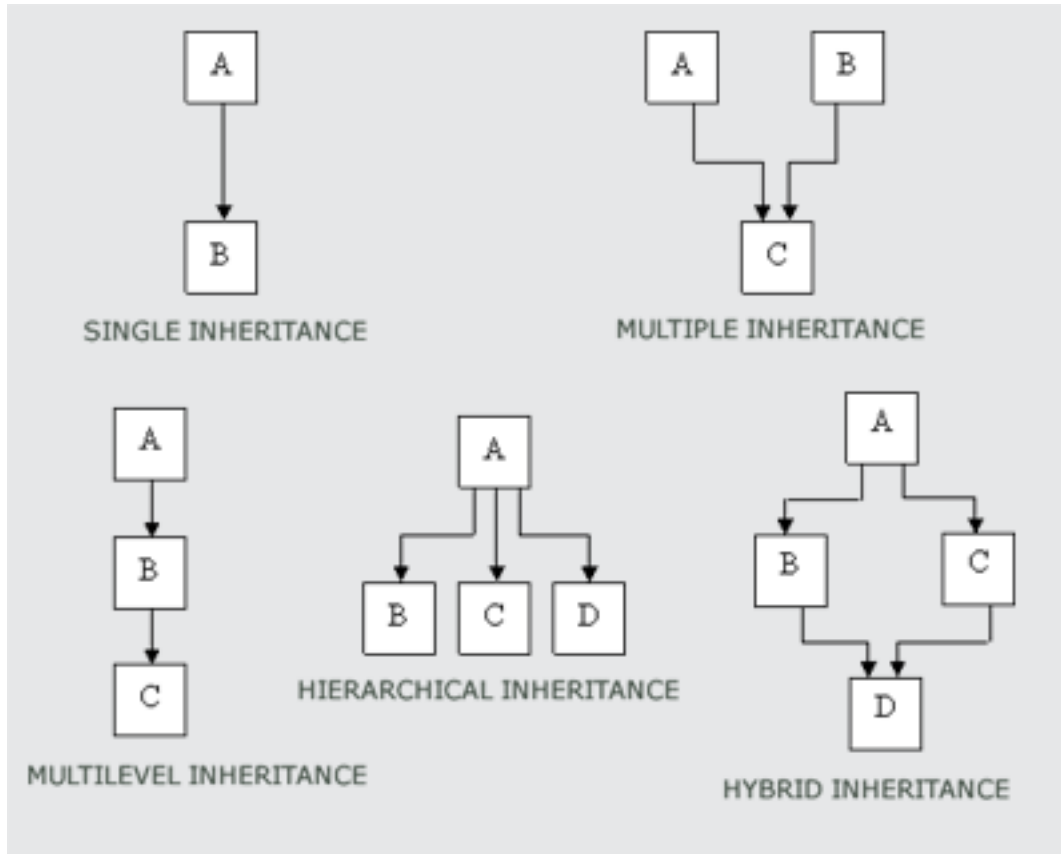
One of the most important concepts in object-oriented programming is that of inheritance. Inheritance allows us to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to reuse the code functionality and fast implementation time. When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the **base** class, and the new class is referred to as the **derived** class. The idea of inheritance implements the **is a** relationship. For example, mammal IS-A animal, dog IS-A mammal hence dog IS-A animal as well and so on.



Types of inheritance:

1. Single Inheritance
2. Multiple Inheritance
3. Multilevel Inheritance
4. Hierarchical Inheritance
5. Hybrid Inheritance

Developed By Strawberry



Syntax:

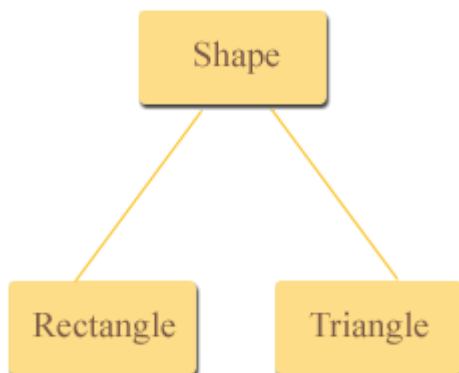
```
class derived_class: memberAccessSpecifier base_class
{
    //code inside...
};
```

Where `derived_class` is the name of the derived class and `base_class` is the name of the class on which it is based. The member Access Specifier may be `public`, `protected` or `private`. This access specifier describes the access level for the members that are inherited from the base class.

Developed By Strawberry

Member Access Specifier	How Members of the Base Class Appear in the Derived Class
Private	Private members of the base class are inaccessible to the derived class.
	Protected members of the base class become private members of the derived class.
	Public members of the base class become private members of the derived class.
Protected	Private members of the base class are inaccessible to the derived class.
	Protected members of the base class become protected members of the derived class.
	Public members of the base class become protected members of the derived class.
Public	Private members of the base class are inaccessible to the derived class.
	Protected members of the base class become protected members of the derived class.
	Public members of the base class become public members of the derived class.

Following example further explains concept of inheritance :



Developed By Strawberry

```

class Shape
{
protected:
    float width, height;
public:
    void set_data (float a, float b)
    {
        width = a;
        height = b;
    }
};

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

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

int main ()
{
    Rectangle rect;
    Triangle tri;
    rect.set_data (5,3);
}

```

Developed By Strawberry

```
    tri.set_data (2,5);  
    cout << rect.area() << endl;  
    cout << tri.area() << endl;  
    return 0;  
}
```

Output:

15

5

Constructors in derived class:

Constructor play important role in initializing objects. One important thing to note here is that as long as no base class constructor takes any arguments, the derived class need not have a constructor function. However if the vase class contains a constructor with one or more arguments, then it is mandatory for derived class to have a constructor and pass the argument to the base class constructor. Remember while applying inheritance, usually objects are created using derived class. Thus it makes sense for the derived class, to pass the argument to the base class constructor..When both derived and base classes contain constructors, the base class constructor is executed first and then the constructor in the derived class is executed.

Example:

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

```
class alpha {  
    int x;  
    public:  
        alpha(int i){  
            x=i;  
            cout<<"alpha initialized \n";  
        }  
}
```

Developed By Strawberry


```

        void show_x(void){
        cout<<"x=" << x << "\n";
        }

};

class beta{
    float y;
    public:
        beta(float j){
            y=j;
            cout<<"beta initialized \n";
        }

        void show_y(void){
        cout<<"y=" << y << "\n";
        }

};

class gamma : public beta, public alpha{
    int m,n;
    public:
        gamma(int a, float b, int c, int d): alpha(a), beta(b)
        {
            m=c;
            n=d;
            cout<<"gamma initialized \n" << "\n";
        }

        void show_mn(void){
        cout<<"m=" << m << "\n"
            <<"n=" << n << "\n";
        }

};

int main(){

```

Developed By Strawberry

```
gamma g(5,10,75,20,30);  
cout<<"\n";  
g.show_x();  
g.show_y();  
g.show_mn();  
return 0;  
}
```

Output:

```
beta initialized  
alpha initialized  
gamma initialized
```

```
x=5  
y=10.75  
m=20  
n=30
```

Developed By Strawberry

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 there is no Blackboard access available)

Roll No. N008	Name: AKSHAY BANDA
Program: MBA TECh CS	Division: C
Semester: 2	Batch : C1
Date of Experiment: 4/3/2015	Date of Submission: 4/3/2015
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 staff
{
protected:
    int c;
    string n;
public:
    void sinput()
    {
        cout<<"\nEnter code : ";
        cin>>c;
        cout<<"\nEnter name : ";
        cin>>n;
    }
    void idisplay()
    {
```

Developed By Strawberry

```

        cout<<"\nCode : "<<c;
        cout<<"\nName : "<<n;
    }
};

class teacher : public staff
{
protected:
    string s;
    string p;
public:
    void tinput()
    {
        sinput();
        cout<<"\nEnter subject : ";
        cin>>s;
        cout<<"\nEnter publication : ";
        cin>>p;
    }
    void tdisplay()
    {
        idisplay();
        cout<<"\nSubject : "<<s;
        cout<<"\nPublication : "<<p;
    }
};

```

```

class officer : public staff
{
protected:
    string g;
public:
    void oinput()
    {
        sinput();
        cout<<"\nEnter grade : ";
        cin>>g;
    }
    void odisplay()
    {

```

Developed By Strawberry

```

        idisplay();
        cout<<"\nGrade : "<<g;
    }
};

class typist : public staff
{
protected:
    double s;
public:
    void tpinput()
    {
        sinput();
        cout<<"\nEnter speed : ";
        cin>>s;
    }
    void tydisplay()
    {
        idisplay();
        cout<<"\nSpeed "<<s;
    }
};

class regular : public typist
{
protected:
    double sal;
public:
    void input()
    {
        tpinput();
        cout<<"\nEnter monthly salary : ";
        cin>>sal;
    }
    void display()
    {
        tydisplay();
        cout<<"\nSalary : "<<sal;
    }
};

```

Developed By Strawberry

```

class causal : public typist
{
protected:
    double sal;
public:
    void input()
    {
        tinput();
        cout<<"\nEnter daily salary : ";
        cin>>sal;
    }
    void display()
    {
        tydisplay();
        cout<<"\nSalary : "<<sal;
    }
};

```

```

int main()
{
    int c,d;
    cout<<"\nEnter 1 for teacher";
    cout<<"\nEnter 2 for typist";
    cout<<"\nEnter 3 for officer";
    cout<<"\nEnter your choice : ";
    cin>>c;
    if(c==1)
    {
        teacher t;
        t.tinput();
        t.tdisplay();
    }
    else if(c==3)
    {
        officer o;
        o.oinput();
        o.odisplay();
    }
    else if(c==2)

```

Developed By Strawberry

```

{
    cout<<"\nEnter 1 for regular";
    cout<<"\nEnter 2 for causal";
    cout<<"\nEnter your choice : ";
    cin>>d;
    if(d==1)
    {
        regular r;
        r.input();
        r.display();
    }
    else if(d==2)
    {
        causal c;
        c.input();
        c.display();
    }
}
return 0;
}

```

2.

```

#include<iostream>
using namespace std;

```

```

class staff
{
protected:
    int c;
    string n;
public:
    void sinput()
    {
        cout<<"\nEnter code : ";
        cin>>c;
        cout<<"\nEnter name : ";
        cin>>n;
    }
    void idisplay()
    {

```

Developed By Strawberry

```

        cout<<"\nCode : "<<c;
        cout<<"\nName : "<<n;
    }
};

class education
{
protected:
    string eq;
    string pq;
public:
    void einput()
    {
        cout<<"\nEnter Educational Qualification : ";
        cin>>eq;
        cout<<"\nEnter Professional Qualification : ";
        cin>>pq;
    }
    void edisplay()
    {
        cout<<"\nEducational Qualification : "<<eq;
        cout<<"\nProfessional Qualification : "<<pq;
    }
};

```

```

class teacher : public staff, public education
{
protected:
    string s;
    string p;
public:
    void tinput()
    {
        sinput();
        cout<<"\nEnter subject : ";
        cin>>s;
        cout<<"\nEnter publication : ";
        cin>>p;
        einput();
    }
}

```

Developed By Strawberry


```

void tdisplay()
{
    idisplay();
    cout<<"\nSubject : "<<s;
    cout<<"\nPublication : "<<p;
    edisplay();
}
};

```

```

class officer : public staff, public education
{
protected:
    string g;
public:
    void oinput()
    {
        sinput();
        cout<<"\nEnter grade : ";
        cin>>g;
        einput();
    }
    void odisplay()
    {
        idisplay();
        cout<<"\nGrade : "<<g;
        edisplay();
    }
};

```

```

class typist : public staff
{
protected:
    double s;
public:
    void tpinput()
    {
        sinput();
        cout<<"\nEnter speed : ";
        cin>>s;
    }
};

```

Developed By Strawberry

```

void tydisplay()
{
    idisplay();
    cout<<"\nSpeed "<<s;
}
};

class regular : public typist
{
protected:
    double sal;
public:
    void input()
    {
        tinput();
        cout<<"\nEnter monthly salary : ";
        cin>>sal;
    }
    void display()
    {
        tydisplay();
        cout<<"\nSalary : "<<sal;
    }
};

```

```

class causal : public typist
{
protected:
    double sal;
public:
    void input()
    {
        tinput();
        cout<<"\nEnter daily salary : ";
        cin>>sal;
    }
    void display()
    {
        tydisplay();
        cout<<"\nSalary : "<<sal;
    }
};

```

Developed By Strawberry

```

    }
};

int main()
{
    int c,d;
    cout<<"\nEnter 1 for teacher";
    cout<<"\nEnter 2 for typist";
    cout<<"\nEnter 3 for officer";
    cout<<"\nEnter your choice : ";
    cin>>c;
    if(c==1)
    {
        teacher t;
        t.tinput();
        t.tdisplay();
    }
    else if(c==3)
    {
        officer o;
        o.oinput();
        o.odisplay();
    }
    else if(c==2)
    {
        cout<<"\nEnter 1 for regular";
        cout<<"\nEnter 2 for causal";
        cout<<"\nEnter your choice : ";
        cin>>d;
        if(d==1)
        {
            regular r;
            r.input();
            r.display();
        }
        else if(d==2)
        {
            causal c;
            c.input();
            c.display();
        }
    }
}

```

Developed By Strawberry

```
    }  
  }  
  return 0;  
}
```

B.2 Input and Output:

(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 teacher
Enter 2 for typist
Enter 3 for officer
Enter your choice : 2

Enter 1 for regular
Enter 2 for casual
Enter your choice : 2

Enter code : 7

Enter name : akshay

Enter speed : 100

Enter daily salary : 10000

Code : 7
Name : akshay
Speed 100
Salary : 10000

2.

Enter 1 for teacher
Enter 2 for typist

Developed By Strawberry

Enter 3 for officer
Enter your choice : 1

Enter code : 7

Enter name : akshay

Enter subject : computers

Enter publication : strawberry

Enter Educational Qualification : MBA

Enter Professional Qualification : Chairman

Code : 7

Name : akshay

Subject : computers

Publication : strawberry

Educational Qualification : MBA

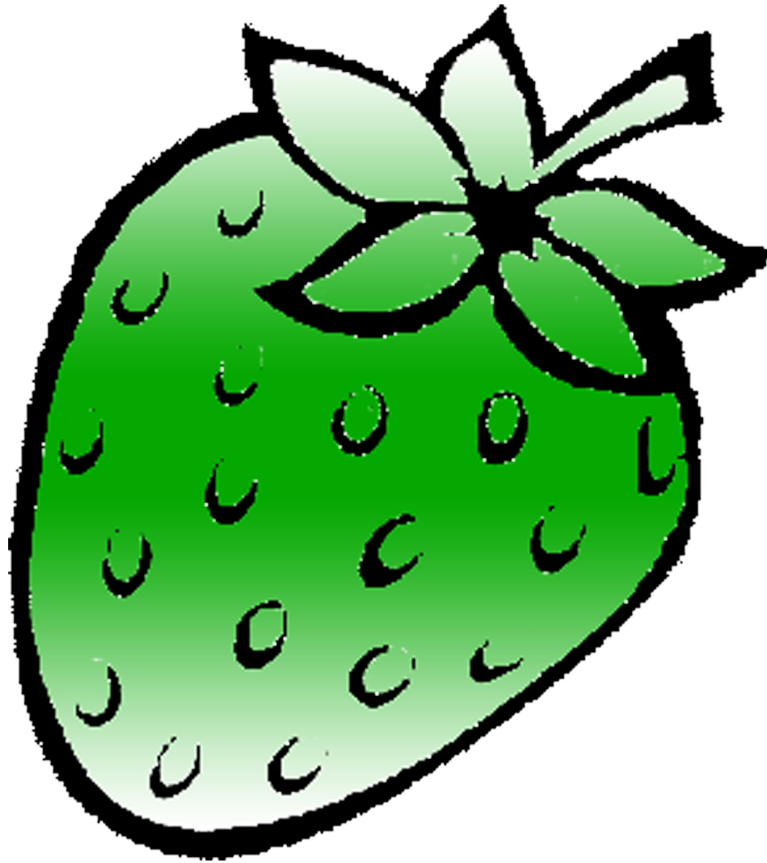
Professional Qualification : Chairman

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)

Developed By Strawberry

STRAWBERRY



/strawberrydevelopers



/strawberry_app

For more visit:

Strawberrydevelopers.weebly.com