

Lab-8

Title: Constructor in Inheritance

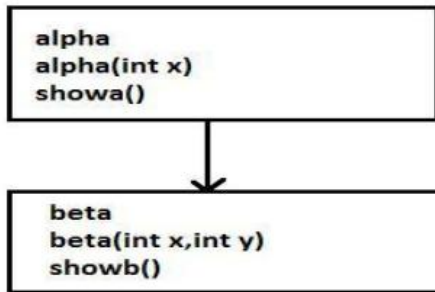
Objective:

To be familiar with use of constructor in inheritance

To understand about working mechanism of constructor and destructor in inheritance.

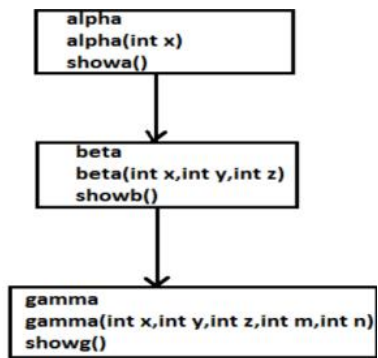
Theory: Constructor and destructor in inheritance Argument passing mechanism for supplying initial values to the bases classes constructors.

1. Write a complete program with reference to given below.



```
#include<iostream>

using namespace std;
class alpha{
    private:
        int a;
    public:
        alpha(int x){
            a=x;
        }
        void showa(){
            cout<<"value of a"<<a<<endl;
        }
};
class beta:public alpha{
    private:
        int b;
    public:
        beta(int x,int y):alpha(x){
            b=y;
        }
        void showb(){
            cout<<"Value of b"<<b<<endl;
        }
};
int main(){
    beta b1(4,5);
    b1.showa();
    b1.showb();
    return 0;
}
```

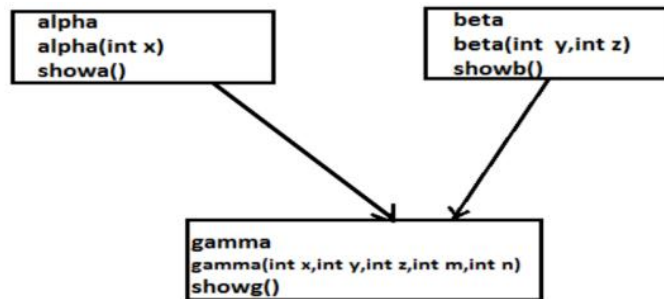


```
#include<iostream>
using namespace std;
class alpha{
private:
    int a;
public:
    alpha(int x){
        a=x;
    }
    void showa(){
        cout<<"value of a:"<<a<<endl;
    }
};
class beta:public alpha{
private:
    int b,c;
public:
    beta(int x,int y,int z):alpha(x){
        b=y;
        c=z;
    }
    void showb(){
        cout<<"Value of b:"<<b<<endl;
        cout<<"Value of c:"<<c<<endl;
    }
};
class gamma:public beta{
private:
    int d,e;
public:
    gamma(int x,int y,int z,int m,int n):beta(x,y,z){
        d=m;
        e=n;
    }
    void showg(){
        cout<<"Value of d:"<<d<<endl;
        cout<<"Value of e:"<<e<<endl;
    }
};
```

```

    }
};
int main(){
    gamma g1(4,5,6,7,8);
    g1.showa();
    g1.showb();
    g1.showg();
    return 0;
}

```



```

#include<iostream>
using namespace std;
class alpha{
    private:
        int a;
    public:
        alpha(int x){
            a=x;
        }
        void showa(){
            cout<<"value of a:"<<a<<endl;
        }
};
class beta{
    private:
        int b,c;
    public:
        beta(int y,int z){
            b=y;
            c=z;
        }
        void showb(){
            cout<<"Value of b:"<<b<<endl;
            cout<<"Value of c:"<<c<<endl;
        }
};

```

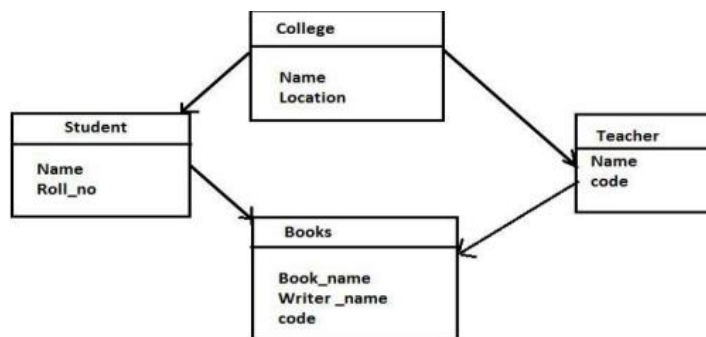
```

class gamma:public alpha,public beta{
    private:
        int d,e;
    public:
        gamma(int x,int y,int z,int m,int n):alpha(x),beta(y,z){
            d=m;
            e=n;
        }
        void showg(){
            cout<<"Value of d:"<<d<<endl;
            cout<<"Value of e:"<<e<<endl;
        }
};

int main(){
    gamma g1(4,5,6,7,8);
    g1.showa();
    g1.showb();
    g1.showg();
    return 0;
}

```

2. The following figure shows the minimum information required for each class. Write a program to realize the above program with necessary member functions to create the database and retrieve individual information .Every class should contain at least one constructor and should be inherited to other classes as well.



```

#include<iostream>
#include<cstring>
using namespace std;
class College{
    protected:
        char Name[50];
        char Location[20];
    public:

```

```

        College(char na[],char l[]){
            strcpy(Name,na);
            strcpy(Location,l);
        }
        void display(){
            cout<<"Name of college:"<<Name<<endl;
            cout<<"Location of college:"<<Location<<endl;
        }
    };

class Student: virtual public College{
protected:
    char Name[20];
    int Roll_no;
public:
    Student(char na[],char l[],char n[],int R):College(na,l){
        strcpy(Name,n);
        Roll_no=R;
    }
    void display(){
        cout<<"Name of student:"<<Name<<endl;
        cout<<"Roll number of student:"<<Roll_no<<endl;
    }
};

class Teacher: virtual public College{
protected:
    char Name[20];
    int code;
public:
    Teacher( char na[],char l[],char tn[],int c):College(na,l){
        strcpy(Name,tn);
        code=c;
    }
    void display(){
        cout<<"Name of teacher:"<<Name<<endl;
        cout<<"COde of teacher:"<<code<<endl;
    }
};

class Books:public Student,public Teacher{
protected:
    char Book_name[30];
    char Writer_name[30];
    int code;
public:
    Books(char na[],char l[],char n[],int R,char tn[],int c,char b[],char w[],int
co):Student(na,l,n,R),Teacher(na,l,tn,c),College(na,l){
        strcpy(Book_name,b);
        strcpy(Writer_name,w);
        code=co;
    }
};

```

```

    }
    void display(){
        cout<<"Name of Book:"<<Book_name<<endl;
        cout<<"Name of Writer:"<<Writer_name<<endl;
        cout<<"COde of Book:"<<code<<endl;
    }
};

int main(){

    Books b1("EEC", "Sanepa", "LerBron", 23, "Jordan", 143, "OOP in c++", "pradip", 33);
    b1.College::display();
    b1.Student::display();
    b1.Teacher::display();
    b1.display();
    return 0;
}

```

3. Create a class person with data members name, age and address. Create another class teacher with data members Qualification and department .Also create another class Student with data members program and semester. Both class are inherited from class person. Every class has at least one constructor which uses base class constructor. Create member function showdata() in each to display the information of the class member.

```

#include<iostream>
#include<cstring>
using namespace std;
class Person{
protected:
    char name[20];
    int age;
    char address[20];
public:
    Person(char n[],int a,char ad[]){
        strcpy(name,n);
        age=a;
        strcpy(address,ad);
    }
    void show_data(){
        cout<<"Name:"<<name<<endl;
        cout<<"Age:"<<age<<endl;
        cout<<"Address:"<<address<<endl;
    }
};

class Teacher:public Person{
protected:
    char qualification[20];
    char department[20];
public:
    Teacher(char n[],int a,char ad[],char q[],char d[]):Person(n,a,ad)
    {

```

```

        strcpy(qualification,q);
        strcpy(department,d);
    }
    void show_data(){
        Person::show_data();
        cout<<"Qualification:"<<qualification<<endl;
        cout<<"Department:"<<department<<endl;
    }
};
class Student:public Person{
protected:
    char program[20];
    char sem[20];
public:
    Student(char n[],int a,char ad[],char p[],char s[]):Person(n,a,ad){
        strcpy(program,p);
        strcpy(sem,s);
    }
    void show_data(){
        Person::show_data();
        cout<<"Program:"<<program<<endl;
        cout<<"Semester:"<<sem<<endl;
    }
};
int main(){
    Teacher t1("Curry",20,"USA","Phd","Civil");
    t1.show_data();
    Student s1("MARCUS",16,"UK","Computer","eighth");
    s1.show_data();
    return 0;
}

```

4. Create two classes Publication and Sales. The Publication class holds title and price and the Sales class holds the total monthly sales. Derive a class called Book from both Publication and Sales. The book class stores author's name and page count. Each of these three classes should have a function Display() to print the information of classes and should have at least one constructor that is inherited to derived class as well. Write a program to implement these classes.

```

#include <iostream>
#include <cstring>
using namespace std;
class Publication {
protected:
    char title[50];
    float price;

public:
    Publication(char t[], float p) {
        strcpy(title, t);
    }
};

```



```

        price=p;
    }
    void Display() {
        cout<<"Title:"<<title<<endl;
        cout<<"Price:"<<price<<endl;
    }
};
class Sales {
protected:
    float sales;

public:
    Sales(float s) {
        sales=s;
    }

    void Display() {
        cout<<"Sales:"<<sales<<endl;
    }
};
class Book:public Publication,public Sales {
private:
    char author[50];
    int page;

public:
    Book(char t[], float p, float s,char a[], int pages): Publication(t, p), Sales(s) {
        strcpy(author, a);
        page=pages;
    }
    void Display(){
        Publication::Display();
        Sales::Display();
        cout<<"Author:"<<author<<endl;
        cout<<"Page Count:"<<page<<endl;
    }
};
int main() {
    Book b1("Muna Madan",100.00,10000, "Laxmi Prassad Devkota",60);
    cout << "Book Information:\n";
    b1.Display();
    return 0;
}

```