```
//  Inheritance

using namespace std;

#include<iostream>

#include<string.h>

struct Employee                        //Base class

{

        int emp_id;

        char name[20];

        double salary;

        Employee()

        {

                cout<<"\n\nEmp default constructor called\n";

                this->emp_id=0;

                strcpy(this->name,"not_given");

                this->salary=0;

        }

        Employee(int i,const char* n,double s)

        {

                cout<<"\n\nEmp parameterised called\n";

                this->emp_id=i;

                strcpy(this->name,n);

                this->salary=s;

        }

        void setId(int i)  //setters(mutators)

        {
                this->emp_id=i;
        }
        void setName(const char* n)     //setters(mutators)
        {
                strcpy(this->name,n);
        }
        void setSalary(double s) //setters(mutators)
        {
                this->salary=s;
        }
        int getId()             //getters(accessors)
        {
                return this->emp_id;
        }
        char* getName()         //getters(accessors)
        {
                return this->name;
        }
        double getSalary()      //getters(accessors)
        {
                return this->salary;
        }
        void display()
        {
                cout<<"\nemployees detail: \nid: "<<this->emp_id<<"\tname: "<<this->name<<"\tsalary: "<<this->salary<<"\n";
        }
};
struct SalesMan:public Employee            //step 1
{
        int target;
        double intensive;
        SalesMan():Employee()  //step 2(a)
        {
                cout<<"\n\nSM default constructor called\n";
                this->target=0;
                this->intensive=0;
        }
        SalesMan(int i,const char* n,double s,int t,int in):Employee(i,n,s)                //step 2(b)
        {
                printf("\n\nSM parameterised constructor called\n");
                this->target=t;
                this->intensive=in;
        }
        void setTarget(int t)                //extra setters(mutator) required for Sales manager
        {
                this->target=t;
        }
        void setIntense(double in)                //extra setters(mutator) required for Sales manager
        {
                this->intensive=in;
        }
        int getTarget()             //extra getters(accessor) required for sales manager
        {
                return this->target;
        }
        double getIntense()                 //extra getters(accessor) required for sales manager
        {
                return this->intensive;
        }
        void display()
        {
                Employee::display();    //step 3
                cout<<"\ntarget: "<<this->target<<"\tintensive: "<<this->intensive;
        }
};
struct Admin:public Employee    //step 1
{
        double allowance;
        Admin():Employee()      //step 2(a)
        {
                cout<<"\n\nAdmin default constructor called\n";
                this->allowance=0;
        }
        Admin(int i,const char* n,double s,double a):Employee(i,n,s)      //step 2(b)
        {
```

```
		cout<<"\n\nAdmin parameterised constructor called\n";
		this->allowance=a;
	}
	void setAllow(double a)		//extra setters(mutator) required for Admin
	{
		this->allowance=a;
	}
	double getAllow()	//extra getters(accessor) required for Admin
	{
		return this->allowance;
	}
	void display()	//step 3
	{
		Employee::display();
		cout<<"\tallowance: "<<this->allowance<<"\n";
	}
};
struct HrManager:public Employee	//step 1
{
	double commission;
	HrManager():Employee()		//step 2(a)
	{
		cout<<"\n\nHR default constructor called\n";
		this->commission=0;
	}
```

```
        HrManager(int i,const char* n,double s,double c):Employee(i,n,s)          //step 2(b)
        {
                cout<<"\n\nHR parameterised constructor called\n";
                this->commission=c;
        }
        void setComm(double c)                    //extra setters(mutator) required for HR manager
        {
                this->commission=c;
        }
        double getComm()       //extra getters(accessor) required for HR manager
        {
                return this->commission;
        }
        void display()
        {
                Employee::display();     //step 3
                cout<<"\tCommission: "<<this->commission<<"\n";
        }
};
int main()
{
        SalesMan m1;
        int id,target;
        char name[20];
        double salary,intensive,allowance,commission;
```

```
    SalesMan m3;
    m3.display();
    SalesMan m2(22,"pragati",50000,45,4500);
    m2.display();
    Admin a1;
    a1.display();
    Admin a2(101,"pragati",50000,4500);
    a2.display();
    HrManager h1;
    h1.display();
    HrManager h2(202,"pragati",50000,5000);
    h2.display();
    return 0;
}
```