1.

```
#include<stdio.h>

#include<string.h>


struct Employee {

        int eID;

        char eName[30];

        float eSallary;


        Employee() {


                eID=0;

                strcpy(eName,"null");

                eSallary=0.0;

        }

        Employee(int eID,char*name,float sallary) {


                this->eID=eID;

                strcpy(this->eName,name);

                this->eSallary=sallary;

        }

        void setEID(int id) {

                this->eID=id;

        }

        void setEName(char*name) {

                strcpy(this->eName,name);


        }

        void setESallary(float sal) {

                this->eSallary=sal;

        }
```

```
	void display() {
		printf("\nid=%d\n",eID);
		printf("Name=%s\n",eName);
		printf("sallary=%.2f\n",eSallary);
	}
	virtual float calsal() {
		printf("Total sallary of Employee=%f",this->eSallary);
		return this->eSallary;
	}
};
struct SalesManager :public Employee {

	int taregt;
	float incentives;

	SalesManager() {

		this->taregt=0;
		this->incentives=0;
	}
	SalesManager(int id,char*name,float sallary,int target,float
incentives):Employee(id,name,sallary) {

		this->taregt=target;
		this->incentives=incentives;
	}
	void setTarget(int a) {
		this->taregt=a;
	}
	void setIncentives(int a) {
		this->incentives=a;
```

```
        }
        void display() {
                Employee::display();
                printf("taregt=%d\n",this->taregt);
                printf("Incentives=%.2f\n",this->incentives);
        }
        virtual float calsal() {
                printf("Total Sallary of Salesmanager=%.2f",this->eSallary+this->incentives);
                return this->eSallary+this->incentives;
        }
};
struct HR :public Employee {

        float cummision;

        HR() {

                this->cummision=0;
        }
        HR(int id,char*name,float sal,float c):Employee(id,name,sal) {

                this->cummision=c;
        }
        void setCummission(float cummision) {
                this->cummision=cummision;
        }
        float getCummision() {
                return this->cummision;
        }
        void display() {
                Employee::display();
```

```
		printf("cummission=%.2f\n",this->cummision);
	}
	float calsal() {
		printf("Total Sallary of HR=%.2f",this->eSallary+this->cummision);
		return this->eSallary+this->cummision;
	}
};
struct Admin :public Employee {

	double allowance;
	Admin() {

		this->allowance=0;
	}
	Admin(int id,char*name,float sallary,double allowance):Employee(id,name,sallary) {

		this->allowance=allowance;
	}
	void setAllowance(double allowance) {
		this->allowance=allowance;
	}
	void display() {
		Employee::display();
		printf("allowance=%.2lf\n",this->allowance);
	}
	float calsal() {
		printf("Total Sallary of Admin=%.2f",this->eSallary+this->allowance);
		return this->eSallary+this->allowance;
	}
};
struct AreaSalesManager :public SalesManager {
    char area[10];
    AreaSalesManager() {

        strcpy(this->area,"null");
    }
    AreaSalesManager(int id,char*name,float sallary,int target,float incentives,char*area):SalesManager(id,name,sallary,target,incentives) {

        strcpy(this->area,area);
    }
    void setArea(char*area) {
        strcpy(this->area,area);
    }
    void display() {
        SalesManager::display();
        printf("Area=%s\n",this->area);
    }
    float calsal() {
        printf("Total Sallary of AreaSalesmanager=%.2f",this->eSallary+this->incentives);
        return this->eSallary+this->incentives;
    }
};
int main() {

    //creating array of pointers
    Employee*p[10];  //---->pointer of Employee to store the addresses of different objects

    p[0]=new SalesManager(101,"xyz",10000.0,153,1000);  //===>p[0]=&s; bacause new returns the pointer of Salesmanager i.d address of SalesManager
    p[1]=new HR(102,"abc",10000.0,2000);
    p[2]=new Admin(103,"pqr",10000,1200);
    p[3]=new AreaSalesManager(104,"yahs",10000,143,1000,"pune");
    float totalSallary=0;
    for(int i=0; i<4; i++) {
        totalSallary+=p[i]->calsal();
        printf("\n");

    }
    printf("\ntotalSallary of All Employee=%.2lf",totalSallary);

}
```

2.

```
#include<stdio.h>

struct Shapes {
        float Area;

    public:

        Shape() {
            this->Area=0.0;
        }
        Shape(float area) {
            this->Area=area;
        }

        virtual float calArea() {
            return this->Area;
        }
        virtual void displayArea() {
            printf("Area=%f\n",this->Area);
        }
};
struct Rectangle :public Shapes {
    float height,length,width;

    Rectangle() {
        this->height=0;
        this->length=0;
        this->width=0;
    }
    Rectangle(int height,int length,int width) {
        this->height=height;
        this->length=length;
        this->width=width;

    }
    float calArea() {
        this->Area=width*length;
        return  Area;
    }
    void displayArea() {
        printf("Area of Rectangle=%f\n",this->Area);
    }
};
struct Triangle :public Shapes {
    float height,base;

    Triangle() {
        this->height=0;
        this->base=0;
        }
        Triangle(int height,int base) {
                this->height=height;
                this->base=base;
        }
        float calArea() {
                this->Area=(this->height*this->base)/2;
                return  Area;
        }
        void displayArea() {
                printf("Area of Triangle=%f\n",this->Area);
        }
};
struct Circle:public Shapes {
        int radius;
        Circle() {
                this->radius=0;
        }
        Circle(int radius) {
                this->radius=radius;
        }
        float calArea() {
                this->Area=radius*radius*3.14;;
                return  Area;
        }
        void displayArea() {
                printf("Area of Circle=%f\n",this->Area);
        }
};
struct Square:public Shapes {
        int side;
        Square() {
                this->side=0;
        }
        Square(int side) {
                this->side=side;
        }
        float calArea() {
                this->Area=this->side*this->side;
                return  Area;
        }
        void displayArea() {
                printf("Area of Square=%f\n",this->Area);
        }
};



int main() {
        Shapes*s[10];
        s[0]=new Circle(10);
        s[1]=new Triangle(5,2);
        s[2]=new Rectangle(10,5,2);
        s[3]=new Square(10);


        float total=0;
        for(int i=0;i<4;i++)
        {
                total+=s[i]->calArea();
                s[i]->displayArea();
        }
```

```
    printf("\nTotal Area of all shapes=%.2f",total);



}
```

3.

```
#include<stdio.h>
#include<string.h>
class Vehicle {

    public:

        virtual void start() {
            printf("vehicle is strarting\n");
        }
};
class Car:public Vehicle {

    public:

        void start() {
            printf("Car is strarting\n");
        }
};
class Cycle:public Vehicle {

    public:

        void start() {
            printf("cycle is strarting\n");
        }
};
class truck:public Vehicle {

    public:

        void start() {
            printf("truck is strarting\n");
        }
};
int main() {

    Vehicle* m[4];

    m[0]=new Car;
    m[1]=new Cycle;
    m[2]=new truck;
    m[3]=new Vehicle;

    for(int i=0; i<4; i++) {
        m[i]->start();
    }

    return 0;
}
```

4.

```
#include<stdio.h>
#include<string.h>
class MusicalInstruments {
    public:

        virtual void display() {
            printf("Instrument is Playing\n");
        }
};
class Guitar:public MusicalInstruments {

    public:

        void display() {
    printf("Guitar is Playing\n");
        }
};
class Piano:public MusicalInstruments {

    public:

        void display() {
        printf("Piano is Playing\n");
        }
};
class Violine:public MusicalInstruments {

    public:

        void display() {
    printf("Violine is Playing\n");
        }
};
```

```
int main() {

   Guitar g;
            MusicalInstruments*p=&g;
            p->display();

    Piano pi;
            MusicalInstruments*p2=&pi;
            p2->display();

    Violine vi;
            MusicalInstruments*p3=&vi;
            p3->display();

            return 0;
}
```

5.

```
#include<stdio.h>
#include<string.h>
class Person {
                        char name[10];
                        int age;
                        char address[20];
            public:
                        Person() {
                                    strcpy(this->name,"null");
                                    this->age=0;
                                    strcpy(this->address,"Null");
                        }
```

```
        Person(char*name,int age,char*address) {
            strcpy(this->name,name);
            this->age=age;
            strcpy(this->address,address);
        }
        virtual void display() {
            printf("Name=%s\n",this->name);
            printf("Age=%d\n",this->age);
            printf("Address=%s\n",this->address);

        }

};
class Student:public Person {
        int studentId;
    public:
        Student() {
            this->studentId=0;
        }
        Student(int studentId,char*name,int age,char*address):Person(name,age,address) {
            this->studentId=studentId;
        }
        void display() {
            Person::display();
            printf("Student Id=%d\n",this->studentId);
            printf("\n*************************\n");
        }
};
class Teacher:public Person {
        int teacherId;
        float sallary;
    public:
        Teacher() {
            this->teacherId=0;
            this->sallary=0.0;
        }
        Teacher(int teacherId,float sallary,char*name,int
age,char*address):Person(name,age,address) {
            this->teacherId=teacherId;
            this->sallary=sallary;
        }
        void display() {
            Person::display();
            printf("Teacher Id=%d\n",this->teacherId);
            printf("Sallary=%.2f",this->sallary);
            printf("\n*************************\n");
        }
};
class Researcher:public Person {
        int Id;

    public:
        Researcher() {
            this->Id=0;
        }
        Researcher(int Id,char*name,int age,char*address):Person(name,age,address) {
            this->Id=Id;
        }
        void display() {
            Person::display();
            printf("Researcher Id=%d\n",this->Id);
            printf("\n*************************\n");
```

```
                }
};
int main() {


        Person*p=new Student(101,"abc",12,"xyz lmn");
        p->display();


        Person* p1=new Teacher(102,1111.11,"xyz",45,"xyz lmn");
        p1->display();


        Person*p2=new Researcher(102,"xyz",45,"xyz lmn");
        p2->display();


        return 0;
}
```