4 assi\_cpp

//bulb

//#include<stdio.h>

#include<iostream>

#include<string.h>

using namespace std;

struct Bulb{

int id;

char cName[50];

double price;

Bulb(){

cout<<"\nDefault constructor of Bulb\n";

this->id=0;

strcpy(this->cName,"Bulb");

this->price=0;

}

Bulb(int i,char\* cnm,double p){

cout<<"\nParameterised constructor of Bulb\n";

this->id=i;

strcpy(this->cName,cnm);

this->price=p;

}

void setId(int id){

this->id=id;

}

void setCname(char\* nm){

strcpy(this->cName,nm);

}

void setPrice(double p){

this->price=p;

}

int getID(){

return this->id;

}

char\* getName(){

return this->cName;

}

double getPrice(){

return this->price;

}

virtual void display(){

cout<<"\nModelId:"<<this->id<<"\n";

cout<<"Company Name:"<<this->cName<<"\n";

cout<<"Price:"<<this->price<<"\n";

}

virtual void toemit()

{

cout<<"Blub is emitting light!!\n";

}

};

struct TugsB:public Bulb{

double volumeTug;

double LenCoil;

TugsB(){

cout<<"\nDefault constructor of TugsB\n";

this->volumeTug=0;

this->LenCoil=0;

}

TugsB(int i,char\* cnm,double p,double vb,double len):Bulb(i,cnm,p){

cout<<"\nparameterised constructor of TugsB\n";

this->volumeTug=vb;

this->LenCoil=len;//melting pints

}

void setVolumetug(double vb)

{

this->volumeTug=vb;

}

void setLenCoil(double len){

this->LenCoil=len;

}

double getVolumetug(){

return this->volumeTug;

}

double getLenCoil(){

return this->LenCoil;

}

void display(){

//use blubs display fun

Bulb::display();

cout<<"volumeTug:"<<this->volumeTug <<"\n";

cout<<"length of coil:"<<this->LenCoil<<"\n";

}

void toemit()

{

cout<<" Tugusten Blub is emitting light!!\n";

}

};//TugsB ends here

struct LED:public Bulb{

double volumeSemC;

LED(){

cout<<"Default constructor of LED\n";

this->volumeSemC=0;

}

LED(int i,char\* Cnm,double p,double vs):Bulb(i,Cnm,p){

cout<<"Parameterised constructor of LED\n";

this->volumeSemC=vs;

}

void SetVolumeS(double vs){

this->volumeSemC=vs;

}

double getVolumeS(){

return this->volumeSemC;

}

void display(){

Bulb::display(); //scope resolution operator

cout<<"volumeSemC:"<<this->volumeSemC<<"\n";

}

void toemit()

{

cout<<" LED Blub is emitting light!!\n";

}

};

int main\_1(){

Bulb b;

b.display();

TugsB t;

TugsB t1(120,"DIP",560,20,12);

t1.display();

LED l;

LED l2(108,"LED",450,62);

l2.display();

return 0;

}

int main()

{

Bulb\* bp;

TugsB t1(120,"DIP",560,20,12);

bp=&t1;

bp->display();

bp->toemit();

LED l2(108,"LED",450,62);

bp=&l2;

bp->display();

bp->toemit();

return 0;

}

//clothes

#include<stdio.h>

#include<string.h>

#include<iostream>

using namespace std;

struct Clothes{

int id;

char clr[50];

char stichBy[50];

double price;

Clothes(){

cout<<"Default constructor of Clothes\n";

this->id=0;

strcpy(this->clr,"Color");

strcpy(this->stichBy,"Prachiti");

this->price=0;

}

Clothes(int id,char\* clr,char\* stich,double p){

cout<<"Parameterisd constructor of Clothes\n";

this->id=id;

strcpy(this->clr,clr);

strcpy(this->stichBy,stich);

this->price=p;

}

void setID(int id){

this->id=id;

}

void setClr(char\* clr){

strcpy(this->clr,clr);

}

void setStichBy(char\* sti){

strcpy(this->stichBy,sti);

}

void setPrice(double p){

this->price=p;

}

int getId(){

return this->id;

}

char\* getClr(){

return this->clr;

}

char\* getStich(){

return this->stichBy;

}

double getPrice(){

return this->price;

}

virtual void display(){

cout<<"ID"<<this->id<<"\n";

cout<<"colour"<<this->clr<<"\n";

cout<<"Stiched by"<<this->stichBy<<"\n";

cout<<"Price:"<<this->price<<"\n";

}

virtual void tostich(){

cout<<"Cloth get stiched by :"<<this->stichBy<<"\n";

}

};

struct Pant:public Clothes{

double waistsize;

double length;

int noOFPackets;

Pant(){

cout<<"Default constructor of Pant!\n";

this->waistsize=0;

this->length=0;

this->noOFPackets=0;

}

Pant(int id,char\* clr ,char\* st,double p,double ws,double l,int pockets):Clothes(id,clr,st,p){

cout<<"Parameterised constructor of Pant!\n";

this->waistsize=ws;

this->length=l;

this->noOFPackets=pockets;

}

void setWaistsize(double ws){

this->waistsize=ws;

}

void setLength(double l){

this->length=l;

}

void setNoOfPockets(int p){

this->noOFPackets=p;

}

double getWaistSize(){

return this->waistsize;

}

double getLength(){

return this->length;

}

int getNoOfPockets(int p){

return this->noOFPackets;

}

void display(){

Clothes::display();

cout<<"waistsize:"<<this->waistsize<<"\n";

cout<<"length:"<<this->length<<"\n";

cout<<"no of pockets:"<<this->noOFPackets<<"\n";

}

void tostich(){

cout<<"Pant get stiched by:"<<this->stichBy <<"and now ready to wear\n";

}

};

struct Tshirt:public Clothes{

double lenSleeves;

double lenShoulder;

Tshirt(){

cout<<"Default constructor of Tshirt !!!\n";

this->lenSleeves=0;

this->lenShoulder=0;

}

Tshirt(int i,char\* clr,char\* st,double p,double sle,double shol):Clothes(i,clr,st,p){

cout<<"Parameterised constructor of Tshirt\n";

this->lenSleeves=sle;

this->lenShoulder=shol;

}

void setLenSle(double sle){

this->lenSleeves=sle;

}

void setLenShol(double shol){

this->lenShoulder=shol;

}

double getLenSle(){

return this->lenSleeves;

}

double getLenShol(){

return this->lenShoulder;

}

void display(){

Clothes::display();

cout<<"Length of sleeves:"<<this->lenSleeves<<"\n";

cout<<"Length of Sholder:"<<this->lenShoulder<<"\n";

}

void tostich(){

cout<<"Tshirt get stiched by "<<this->stichBy<< "and now ready to wear\n";

}

};

int main\_1(){

Pant p1(101,"Pink","Prachiti",5000,32,80,2);

p1.display();

Tshirt t1(102,"black","Prachiti",1000,45,56);

t1.display();

return 0;

}

int main(){

Clothes\* cp;

Pant p1(101,"Pink","Prachiti",5000,32,80,2);

cp=&p1;

cp->display();

cp->tostich();

Tshirt t1(102,"black","Hrutu",1000,45,56);

cp=&t1;

cp->display();

cp->tostich();

return 0;

}

//defeance

#include<stdio.h>

#include<string.h>

#include<iostream>

using namespace std;

struct Defence{

int officerID;

char name[50];

double salary;

Defence(){

cout<<"Default constructor of defence!!\n";

this->officerID=0;

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

this->salary=0;

}

Defence(int id,char\* nm,double s){

cout<<"parameterised constructor of defence!!\n";

this->officerID=id;

strcpy(this->name,nm);

this->salary=s;

}

void setOfficerID(int id){

this->officerID=id;

}

void setName(char\* nm){

strcpy(this->name,nm);

}

void setSalary(double s){

this->salary=s;

}

int getID(){

return this->officerID;

}

char\* getName(){

return this->name;

}

double getSalary(){

return this->salary;

}

void display()

{

cout<<"Officer ID:"<<this->officerID<<"\n";

cout<<"Officer Name:"<<this->name<<"\n";

cout<<"Salary"<<this->salary<<"\n";

}

};

struct Army:public Defence{

int guns;

int tanks;

Army():Defence(){

cout<<"Default constructor of Army !!\n";

this->guns=0;

this->tanks=0;

}

Army(int id,char\* nm,double s,int g,int t):Defence(id,nm,s){

cout<<"Default constructor of Army !!\n";

this->guns=g;

this->tanks=t;

}

void setGuns(int g){

this->guns=g;

}

void setTanks(int t){

this->tanks=t;

}

int getGuns(){

return this->guns;

}

int getTanks(){

return this->tanks;

}

void display(){

Defence::display();

cout<<"No of guns: "<<this->guns<<"\n";

cout<<"No of Tanks: "<<this->tanks<<"\n";

}

};

struct Airforce:public Defence{

int jets;

int helicopter;

Airforce():Defence(){

cout<<"Default constructor of Airforce!!\n";

this->jets=0;

this->helicopter=0;

}

Airforce(int id,char\* nm,double s,int j,int h):Defence(id,nm,s){

cout<<"Parameterised constructor of Airforce!!\n";

this->jets=j;

this->helicopter=h;

}

void setjets(int j){

this->jets=j;

}

void setHeplicopter(int h){

this->helicopter=h;

}

int getJets(){

return this->jets;

}

int getHeplicopter(){

return this->helicopter;

}

void display(){

Defence::display();

cout<<"No of Jets: "<<this->jets<<"\n";

cout<<"No of Heplicopter:"<<this->helicopter<<"\n";

}

};

struct Navy:public Defence{

int ships;

int submarine;

Navy(){

cout<<"Default constructor of Navy!!\n";

this->ships=0;

this->submarine=0;

}

Navy(int id,char\* nm,double s,int ships,int sub):Defence(id,nm,s){

cout<<"Parameterised constructor of Navy!!\n";

this->ships=ships;

this->submarine=sub;

}

void setShips(int ship){

this->ships=ship;

}

void setSubmarine(int sub){

this->submarine=sub;

}

int getShips(){

return this->ships;

}

int getSubmarine(){

return this->submarine;

}

void display(){

Defence::display();

cout<<"No of Ships:"<<this->ships<<"\n";

cout<<"No of Submarine:"<<this->submarine<<"\n";

}

};

int main(){

Army a1(1,"Prachiti",50000,2,4);

a1.display();

Airforce air1(102,"sayali",8000,5,6);

air1.display();

Navy n1(103,"Dip",4500,5,9);

n1.display();

return 0;

}

// melloc calloc realloc strings builtin functions as user defines all, difference betwee while and do while , for loop while loop , assignment question all pointr advantage disadvant

// pointer to structure

//lighter

#include<stdio.h>

#include<string.h>

#include<iostream>

using namespace std;

struct Lighter{

int id;

char Cname[50];

double price;

Lighter(){

cout<<"Default constructor of Lighter called\n";

this->id=0;

strcpy(this->Cname,"Lighter");

this->price=0;

}

Lighter(int i,char\* cnm,double p){

cout<<"Parameterised constructor of Lighter called\n";

this->id=i;

strcpy(this->Cname,cnm);

this->price=p;

}

void setId(int id){

this->id=id;

}

void setCname(char\* nm){

strcpy(this->Cname,nm);

}

void setPrice(double p){

this->price=p;

}

int getID(){

return this->id;

}

char\* getName(){

return this->Cname;

}

double getPrice(){

return this->price;

}

virtual void display(){

cout<<"\nModelId:"<<this->id<<"\n";

cout<<"Company Name:"<<this->Cname<<"\n";

cout<<"Price:"<<this->price<<"\n";

}

virtual void toignit(){

cout<<"Lighter is ignit\n";

}

};

struct FlameL:public Lighter{

//double TankC;//use to store the compressed liquid

double CompLiquid;//volume

FlameL(){

cout<<"Default constructor of FlameL\n";

this->CompLiquid=0;

}

FlameL(int i,char\* cnm,double p,double cl):Lighter(i,cnm,p){

cout<<"Parameterised Constructor of FlameL\n";

this->CompLiquid=cl;

}

void setCompLiquid(double Cl){

this->CompLiquid=Cl;

}

double getCompLiquid(){

return this->CompLiquid;

}

void display(){

//

Lighter::display();

cout<<"Compressed liquid Quantity:"<<this->CompLiquid<<"\n";

}

void toignit(){

printf("Flame Lighter is ignit\n");

}

};//flameL ends here

struct EletricL:public Lighter{

double Battery;

EletricL(){

cout<<"Default constructor of EletricL\n";

this->Battery=0;

}

EletricL(int i,char\* cnm,double p,double b):Lighter(i,cnm,p){

cout<<"Parameterised constructor of EletricL\n";

this->Battery=b;

}

void SetBattery(double b){

this->Battery=b;

}

double getBattery(){

return this->Battery;

}

void display(){

Lighter::display();

cout<<"Battery:"<<this->Battery<<"\n";

}

void toignit(){

cout<<"electric Lighter is ignit\n";

}

};//electric lighter ends here

int main\_1(){

Lighter l;

Lighter l2(102,"Prachiti",500);

l2.display();

FlameL f1;

FlameL f2(105,"Flame",450,30);

f2.display();

EletricL e1;

EletricL e2(106,"hrutu",800,600);

e2.display();

return 0;

}

int main(){

Lighter\* l;

FlameL f2(105,"Flame",450,30);

l=&f2;

l->display();

l->toignit();

EletricL e2(106,"hrutu",800,600);

l=&e2;

l->display();

l->toignit();

return 0;

}

//lock

#include<stdio.h>

#include<string.h>

#include<iostream>

using namespace std;

struct Lock{

int id;

char Cname[40];

char shape[40];

double price;

Lock(){

cout<<"Default constructor of Lock!!\n";

this->id=0;

strcpy(this->Cname,"Cname");

strcpy(this->shape,"circle");

this->price=0;

}

Lock(int i,char\* cn,char\* sp,double p){

cout<<"parameterised constructor of Lock!!\n";

this->id=i;

strcpy(this->Cname,cn);

strcpy(this->shape,sp);

this->price=p;

}

void setId(int i){

this->id=i;

}

void setCname(char\* cn){

strcpy(this->Cname,cn);

}

void setShape(char\* sp){

strcpy(this->shape,sp);

}

void setPrice(double p){

this->price=p;

}

int getId(){

return this->id;

}

char\* getCname(){

return this->Cname;

}

char\* getShape(){

return this->shape;

}

double getPrice(){

return this->price;

}

virtual void display(){

cout<<"Id:"<<this->id<<"\n";

cout<<"Companyname:"<<this->Cname<<"\n";

cout<<"Shape:"<<this->shape<<"\n";

cout<<"Price:"<<this->price<<"\n";

}

virtual void tolock(){

cout<<"Lock get locked!!\n";

}

};

struct DiscLock:public Lock{

int noOfDisc;

DiscLock(){

cout<<"Default constructor of Disclock !!\n";

this->noOfDisc=0;

}

DiscLock(int id ,char\* cn,char\* sp,double p,int d):Lock(id,cn,sp,p){

cout<<"Parameterised constructor of Disclock !!\n";

this->noOfDisc=d;

}

void setDisc(int disc){

this->noOfDisc=disc;

}

double getDisc(){

return this->noOfDisc;

}

void display(){

Lock::display();

cout<<"No of disc "<<this->noOfDisc<<"\n";

}

void tolock(){

cout<<"DiscLock get locked!!\n";

}

};

struct Knob:public Lock{

char materialKnob[40];

Knob(){

printf("Default constructor of Disclock !!\n");

strcpy(this->materialKnob,"steel");

}

Knob(int id ,char\* cn,char\* sp,double p,char\* mk):Lock(id,cn,sp,p){

cout<<"Default constructor of Disclock !!\n";

strcpy(this->materialKnob,mk);

}

void setMknob(char\* mk){

strcpy(this->materialKnob,mk);

}

char\* getMknob(){

return this->materialKnob;

}

void display(){

Lock::display();

cout<<"Material of knob :"<<this->materialKnob<<"\n";

}

void tolock(){

cout<<"knobLock get locked!!\n";

}

};

int main(){

Lock\* lp;

DiscLock d1(101,"Abc","circle",900,8);

lp=&d1;

lp->display();

lp->tolock();

Knob k1(102,"xyz","square",500,"steel");

lp=&k1;

lp->display();

lp->tolock();

return 0;

}

//MIC

//#include<stdio.h>

#include<string.h>

#include<iostream>

using namespace std;

struct Mic{

int id;

char Cname[50];

double price;

Mic(){

cout<<"Default constructor of Mic\n";

this->id=0;

strcpy(this->Cname,"MIC");

this->price=0;

}

Mic(int i,char\* nm,double p){

cout<<"Parameterised constructor of MIC\n";

this->id=i;

strcpy(this->Cname,nm);

this->price=p;

}

void setID(int i){

this->id=i;

}

void setCname(char\* nm){

strcpy(this->Cname,nm);

}

void setPrice(double p){

this->price=p;

}

int getID(){

return this->id;

}

char\* getName(){

return this->Cname;

}

double getPrice(){

return this->price;

}

virtual void display(){

cout<<"ID:"<<this->id<<"\n";

cout<<"Name:"<<this->Cname<<"\n";

cout<<"Price:"<<this->price<<"\n";

}

virtual void toconnect(){

cout<<"Mic is connected!!\n";

}

}; //mic ends

struct WiredMic:public Mic{

char type[60];

WiredMic(){

cout<<"Default constructor wired called\n";

strcpy(this->type,"CType");

}

WiredMic(int i,char\* cnm,double p,char\* t):Mic(i,cnm,p){

cout<<"Parameterised constructor wired called\n";

strcpy(this->type,t);

}

void setType(char\* t){

strcpy(this->type,t);

}

char\* getType(){

return this->type;

}

void display(){

Mic::display();

cout<<"Type :"<<this->type<<"\n";

}

void toconnect(){

cout<<"Mic is connected by wired :"<<this->type<<"\n";

}

};

struct WirelessMic:public Mic{

char versionB[50];

WirelessMic(){

cout<<"default construtor of wireless \n";

strcpy(this->versionB,"Version");

}

WirelessMic(int i,char\*cnm,double p,char\* vb):Mic(i,cnm,p){

cout<<"default construtor of wireless \n";

strcpy(this->versionB,vb);

}

void setVersion(char\* vb){

strcpy(this->versionB,vb);

}

char\* getVersion(){

return this->versionB;

}

void display(){

Mic::display();

cout<<"VersionType:"<<this->versionB<<"\n";

}

void toconnect (){

cout<<"Mic is connected by wirelessly by Bluetooth "<<this->versionB<<"\n";

}

};

int main\_1(){

Mic m;

WiredMic m1(1,"Prachiti",500,"Btype");

m1.display();

WirelessMic w1(2,"Hrutu",677,"version3.4");

w1.display();

return 0;

}

int main()

{

Mic\* mp;

WiredMic m1(1,"Prachiti",500,"Btype");

mp=&m1;

mp->display();

mp->toconnect();

WirelessMic w1(2,"Hrutu",677,"version3.4");

mp=&w1;

mp->display();

mp->toconnect();

return 0;

}

//mirrror

//#include<stdio.h>

#include<string.h>

#include<iostream>

using namespace std;

struct Mirror{

int id;

char shape[20];

char Cname[20];

double price;

Mirror(){

cout<<"Default constructor of Mirror!1\n";

this->id=0;

strcpy(this->shape,"Circle");

strcpy(this->Cname,"Abc");

this->price=0;

}

Mirror(int i,char\* sp,char\* cn,double d){

cout<<"Parameterised constructor of Mirror!1\n";

this->id=i;

strcpy(this->shape,sp);

strcpy(this->Cname,cn);

this->price=d;

}

void setId(int i){

this->id=i;

}

void setShape(char\* sp){

strcpy(this->shape,sp);

}

void setName(char\* cn){

strcpy(this->Cname,cn);

}

void setPrice(double d){

this->price=d;

}

int getId(){

return this->id;

}

char\* getShape(){

return this->shape;

}

char\* getCname(){

return this->Cname;

}

double getPrice(){

return this->price;

}

virtual void display(){

cout<<"ID:"<<this->id<<"\n";

cout<<"Shape:"<<this->shape<<"\n";

cout<<"Company Name:"<<this->Cname<<"\n";

cout<<"Price:"<<this->price<<"\n";

}

virtual void toshow(){

cout<<"Mirror!!\n";

}

};

struct Convex:public Mirror{

Convex(){

cout<<"default constructor of Convex Mirror!!n";

}

Convex(int i,char\* sp,char\* cn,double d):Mirror(i,sp,cn,d){

cout<<"Parameterised constructor of Convex Mirror!!\n";

}

void display(){

Mirror::display();

}

virtual void toshow(){

cout<<"Convex Mirror!!\n";

}

};

struct Concave:public Mirror{

Concave(){

cout<<"default constructor of Concave Mirror!!n";

}

Concave(int i,char\* sp,char\* cn,double d):Mirror(i,sp,cn,d){

cout<<"Parameterised constructor of Concave Mirror!!\n";

}

void display(){

Mirror::display();

}

virtual void toshow(){

cout<<"Concave Mirror!!\n";

}

};

int main(){

Mirror\* mp;

Convex c1(101,"circle","Abc",5000);

mp=&c1;

mp->display();

mp->toshow();

Concave c2(101,"circle","Abc",5000);

mp=&c2;

mp->display();

mp->toshow();

return 0;

}

//phone

#include<stdio.h>

#include<string.h>

#include<iostream>

using namespace std;

struct phone

{

int id;

char CName[40];

double price;

phone(){

cout<<"Default constructor of phone\n";

this->id=0;

strcpy(this->CName,"vivo");

this->price=0;

}

phone(int i,char\* cn,double p){

cout<<"Parameterised constructor of phone\n";

this->id=i;

strcpy(this->CName,cn);

this->price=p;

}

void setId(int id){

this->id=id;

}

void setCname(char\* cn){

strcpy(this->CName,cn);

}

void setPrice(double p){

this->price=p;

}

int getId()

{

return this->id;

}

char\* getCname(){

return this->CName;

}

double getPrice(){

return this->price;

}

virtual void display(){

cout<<"ID:"<<this->id<<"\n";

cout<<"Company Name:"<<this->CName<<"\n";

cout<<"Price:"<<this->price<<"\n";

}

virtual void toCall(){

cout<<"phone is calling\n";

}

};

struct Landline:public phone{

int noKeys;

Landline(){

cout<<"Default constructor of landline \n";

this->noKeys=0;

}

Landline(int i,char\* cn,double p,int k):phone(i,cn,p){

cout<<"Parameterised constructor of landline \n";

this->noKeys=k;

}

void setKeys(int k){

this->noKeys=k;

}

int getKeys(){

return this->noKeys;

}

void display(){

phone::display();

cout<<"no of keys:"<<this->noKeys<<"\n";

}

void tocall(){

cout<<"Landline is calling\n";

}

};

struct smartphone:public phone{

int noSim;

smartphone(){

cout<<"Default constructor of smartphone \n";

this->noSim=0;

}

smartphone(int i,char\* cn,double p,int s):phone(i,cn,p){

cout<<"Parameterised constructor of smartphone \n";

this->noSim=s;

}

void setnoSim(int s){

this->noSim=s;

}

int getnosim(){

return this->noSim;

}

void display(){

phone::display();

cout<<"No of sim:"<<this->noSim<<"\n";

}

void tocall(){

cout<<"smartphone is calling\n";

}

};

int main(){

phone\* p;

Landline l1(101,"vivo",80000,45);

p=&l1;

p->display();

p->toCall();

smartphone s1(102,"samsung",70000,2);

p=&s1;

p->display();

p->toCall();

return 0;

}

//player

#include<stdio.h>

#include<string.h>

#include<iostream>

using namespace std;

struct player

{

int id;

char name[60];

int noTrophies;

player(){

cout<<"Default constructor of player!\n";

this->id=0;

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

this->noTrophies=0;

}

player(int id,char\* nm,int t){

cout<<"parameterised constructor of player!\n";

this->id=id;

strcpy(this->name,nm);

this->noTrophies=t;

}

void setId(int i){

this->id=i;

}

void setName(char\* nm){

strcpy(this->name,nm);

}

void setTrophies(int t){

this->noTrophies=t;

}

int getId(){

return this->id;

}

char\* getName(){

return this->name;

}

double getTrophies(){

return this->noTrophies;

}

virtual void display(){

cout<<"Id:"<<this->id<<"\n";

cout<<"Name:"<<this->name<<"\n";

cout<<"No of trophies:"<<this->noTrophies<<"\n";

}

virtual void toplay(){

cout<<"Player is playing\n";

}

};

struct CricketP:public player{

int noOfwickets;

int noOfRuns;

CricketP(){

cout<<"default constructor called of Cricet player\n";

this->noOfwickets=0;

this->noTrophies=0;

}

CricketP(int i,char\* nm, int Tro,int w,int r):player(i,nm,Tro){

cout<<"Parameterised constructor of cricket player called\n";

this->noOfwickets=w;

this->noOfRuns=r;

}

void setWicket(int w){

this->noOfwickets=w;

}

void setRuns(int r){

this->noOfRuns=r;

}

int getWickets(){

return this->noOfwickets;

}

int getRuns(){

return this->noOfRuns;

}

void display(){

player::display();

cout<<"No of wickets:"<<this->noOfwickets<<"\n";

cout<<"No of Runs :"<<this->noOfRuns<<"\n";

}

void toplay(){

cout<<"Cricket Player is playing cricket\n";

}

};

struct FootballP:public player{

int noOFGoals;

FootballP(){

cout<<"FootBall default constructor called!\n";

this->noOFGoals=0;

}

FootballP(int i,char\* nm,int t,int g):player(i,nm,t){

cout<<"FootBall Parameterised constructor called!\n";

this->noOFGoals=g;

}

void setGoals(int g){

this->noOFGoals=g;

}

int getGoals(){

return this->noOFGoals;

}

void display(){

player::display();

cout<<"No of goals:"<<this->noOFGoals<<"\n";

}

void toplay(){

cout<<"Football Player is playing football\n";

}

};

int main\_1(){

player p;

CricketP c1(101,"Prachiti",5,67,90);

c1.display();

FootballP f1(102,"hrutu",7,80);

f1.display();

return 0;

}

int main()

{

player\* p;

CricketP c1(101,"Prachiti",5,67,90);

p=&c1;

p->display();

p->toplay();

FootballP f1(102,"hrutu",7,80);

p=&f1;

p->display();

p->toplay();

return 0;

}

//teacher

//#include<stdio.h>

#include<string.h>

#include<iostream>

using namespace std;

struct Teacher{

int id;

char name[40];

double Salary;

char Quali[40];

Teacher(){

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

this->id=0;

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

this->Salary=0;

strcpy(this->Quali,"Qualification");

}

Teacher(int i,char\* nm,double s,char\* q){

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

this->id=i;

strcpy(this->name,nm);

this->Salary=s;

strcpy(this->Quali,q);

}

void setId(int id){

this->id=id;

}

void setName(char\* nm){

strcpy(this->name,nm);

}

void setSalary(double s){

this->Salary=s;

}

void setQuali(char\* Qu){

strcpy(this->Quali,Qu);

}

int getID(){

return this->id;

}

char\* getName(){

return this->name;

}

double getSalary(){

return this->Salary;

}

char\* getQu(){

return this->Quali;

}

virtual void display(){

cout<<"ID:"<<this->id<<"\n";

cout<<"name:"<<this->name<<"\n";

cout<<"Salary:"<<this->Salary<<"\n";

cout<<"Qualification:"<<this->Quali<<"\n";

}

virtual void toteach(){

cout<<"Teacher is teaching\n";

}

};

struct DanceT:public Teacher{

int noDance;

int Trophies;

DanceT(){

cout<<"default Constructor of Dancet\n";

this->noDance=0;

this->Trophies=0;

}

DanceT(int i,char\* nm,double p,char\* q,int nd,int t):Teacher(i,nm,p,q){

cout<<"default Constructor of Dancet\n";

this->noDance=nd;

this->Trophies=t;

}

void setNoDance(int nd){

this->noDance=nd;

}

void setTrophies(int t){

this->Trophies=t;

}

int getNoDance(){

return this->noDance;

}

int getTrophies(){

return this->Trophies;

}

void display(){

Teacher::display();

cout<<"No of Dance Known:"<<this->noDance<<"\n";

cout<<"No of trophies:"<<this->Trophies<<"\n";

}

void toteach(){

cout<<"Teacher is teaching Dance\n";

}

};

struct codingT:public Teacher{

int noLang;

int ContestWin;

codingT(){

cout<<"Default constructor of CodingTeacher\n";

this->noLang=0;

this->ContestWin=0;

}

codingT(int i,char\* nm,double p,char\* q,int nl,int cw):Teacher(i,nm,p,q){

cout<<"Parameterised constructor of coding Teacher\n";

this->noLang=nl;

this->ContestWin=cw;

}

void setNoLang(int l){

this->noLang=l;

}

void setContestWin(int c){

this->ContestWin=c;

}

int getNoLang(){

return this->noLang;

}

int getContestWin(){

return this->ContestWin;

}

void display(){

Teacher::display();

cout<<"No of languages known:"<<this->noLang<<"\n";

cout<<"No of contest Win:"<<this->ContestWin<<"\n";

}

void toteach(){

cout<<"Teacher is teaching Coding\n";

}

};

int main\_1(){

Teacher t;

DanceT d1;

DanceT d2(120,"dip",2300,"BA.Dance",3,21);

d2.display();

return 0;

}

int main(){

Teacher\* tp;

DanceT d2(120,"dip",2300,"BA.Dance",3,21);

tp=&d2;

tp->display();

tp->toteach();

codingT c(104,"Prashi",8900,"B.tech",9,10);

tp=&c;

tp->display();

tp->toteach();

return 0;

}

#include"emp.h"

Employee::Employee(){

}

Employee::Employee(int i,char\* nm,double s){

this->id=i;

strcpy(this->name,nm);

this->salary=s;

}

void Employee::setid(int i){

this->id=i;

}

void Employee::setname(char\*nm){

strcpy(this->name,nm);

}

void Employee::setsalary(double s){

this->salary=s;

}

int Employee::getid(){

return this->id;

}

char\* Employee::getname(){

return this->name;

}

double Employee::getsalary(){

return this->salary;

}

void Employee::display(){

cout<<"Employee:\n";

cout<<"id:"<<this->id<<"\n";

cout<<"name:"<<this->name<<"\n";

cout<<"salary:"<<this->salary<<"\n";

}

ostream& operator<<(ostream& o,Employee& e){

o<<"Employer;\n";

o<<"id:"<<e.getid()<<"\n";

o<<"name:"<<e.getname()<<"\n";

o<<"salary:"<<e.getsalary()<<"\n";

o<<"---------------------------------------\n"

return o;

}