OOPS Assignment

By P.V.Srivathsa

22 September 2011

2nd Year CSE “B”

1. ***Write an inheritance hierarchy for class Quadrilateral, Trapezium, Parallelogram, Rectangle and Square. Using Quadrilateral as the base class of the hierarchy, make the hierarchy as deep as possible. The private data of Quadrilateral should be the (x, y) co-ordinate pairs for the 4 points of the Quadrilateral. It also contains member function to calculate the perimeter.***

#include<iostream.h>

#include<math.h>

class quadrilateral

{

protected:

int x1,y1,x2,y2,x3,y3,x4,y4;

public:

quadrilateral()

{}

quadrilateral(int a1,int b1,int a2,int b2,int a3,int b3,int a4,int b4)

{

x1=a1;

x2=a2;

x3=a3;

x4=a4;

y1=b1;

y2=b2;

y3=b3;

y4=b4;

}

float perimeter()

{

float a = sqrt((x2-x1)\*(x2-x1)+(y2-y1)\*(y2-y1));

float b = sqrt((x3-x2)\*(x3-x2)+(y3-y2)\*(y3-y2));

float c = sqrt((x4-x3)\*(x4-x3)+(y4-y3)\*(y4-y3));

float d = sqrt((x1-x4)\*(x1-x4)+(y1-y4)\*(y1-y4));

return a + b + c + d;

}

};

class trapezium: virtual public quadrilateral

{

public:

trapezium()

{

cout << "A Trapezium is a quadrilateral with non parellel sides having equal length" << endl;

}

trapezium(int a1,int b1,int a2,int b2,int a3,int b3,int a4,int b4)

{

cout << "A Trapezium is a quadrilateral with non parellel sides having equal length" << endl;

x1=a1;

x2=a2;

x3=a3;

x4=a4;

y1=b1;

y2=b2;

y3=b3;

y4=b4;

}

};

class plogram: virtual public quadrilateral

{

public:

plogram()

{

cout << "A Parellelogram is a quadrilateral with opposites sides parellel to each other" << endl;

}

plogram(int a1,int b1,int a2,int b2,int a3,int b3,int a4,int b4)

{

cout << "A Parellelogram is a quadrilateral with opposites sides parellel to each other" << endl;

x1=a1;

x2=a2;

x3=a3;

x4=a4;

y1=b1;

y2=b2;

y3=b3;

y4=b4;

}

};

class rect:virtual public quadrilateral,virtual public plogram

{

public:

rect()

{

cout << "A Rectangle is in general a quadrilateral and specifically a parallelogram with all right angles" << endl;

}

rect(int a1,int b1,int a2,int b2,int a3,int b3,int a4,int b4)

{

cout << "A Rectangle is in general a quadrilateral and specifically a parallelogram with all right angles" << endl;

x1=a1;

x2=a2;

x3=a3;

x4=a4;

y1=b1;

y2=b2;

y3=b3;

y4=b4;

}

};

class square:virtual public quadrilateral,virtual public rect

{

public:

square()

{

cout << "A Square is in general a quadrilateral and specifically a rectangle with equal sides" << endl;

}

square(int a1,int b1,int a2,int b2,int a3,int b3,int a4,int b4)

{

cout << "A Square is in general a quadrilateral and specifically a rectangle with equal sides" << endl;

x1=a1;

x2=a2;

x3=a3;

x4=a4;

y1=b1;

y2=b2;

y3=b3;

y4=b4;

}

};

int main()

{

cout << "Initializing quadrilateral" << endl << endl;

quadrilateral q(3,5,-12,3,1,1,-1,0);

cout << endl << endl << "The perimeter is " << q.perimeter() << endl << endl;

cout << "Initializing trapezium" << endl << endl;

trapezium t(-1,0,0,1,2,5,-2,3);

cout << endl << endl << "The perimeter is " << t.perimeter() << endl << endl;

cout << "Initializing parellelogram" << endl << endl;

plogram p(1,4,-4,-3,2,1,0,0);

cout << endl << endl << "The perimeter is " << p.perimeter() << endl << endl;

cout << "Initializing rectangle" << endl << endl;

rect r(1,8,-2,3,-1,4,1,1);

cout << endl << endl << "The perimeter is " << r.perimeter() << endl << endl;

cout << "Initializing square" << endl << endl;

square s(1,1,4,5,6,7,-1,-8);

cout << endl << endl << "The perimeter is " << s.perimeter() << endl << endl;

}

**OUTPUT:**

Initializing quadrilateral

The perimeter is 36.9249

Initializing trapezium

A Trapezium is a quadrilateral with non parellel sides having equal length

The perimeter is 13.5208

Initializing parellelogram

A Parellelogram is a quadrilateral with opposites sides parellel to each other

The perimeter is 22.1726

Initializing rectangle

A Parellelogram is a quadrilateral with opposites sides parellel to each other

A Rectangle is in general a quadrilateral and specifically a parallelogram with

all right angles

The perimeter is 17.8507

Initializing square

A Parellelogram is a quadrilateral with opposites sides parellel to each other

A Rectangle is in general a quadrilateral and specifically a parallelogram with

all right angles

A Square is in general a quadrilateral and specifically a rectangle with equal s

ides

The perimeter is 33.6009

1. ***Write a CPP program that takes two values of time (hr,min,sec) and outputs their sum using constant and operator overloading.***

#include<iostream.h>

class mytime

{

int hrs,min,sec;

static const int xtime = 60;

public:

mytime()

{}

mytime(int h,int m,int s)

{

hrs = h;

if (m > xtime)

{

min = m % xtime;

hrs += (int)(m/xtime);

}

else {

min = m;

}

if(s > xtime)

{

sec = s % xtime;

min += (int)(s/xtime);

}

else

{

sec = s;

}

}

void display()

{

cout << endl << hrs << " : " << min << " : " << sec << endl;

}

mytime operator +(mytime t1)

{

mytime tout;

tout.hrs = t1.hrs + hrs;

int m = t1.min + min;

if (m>xtime)

{

tout.min = m % xtime;

tout.hrs += (int)(m/xtime);

}

else {

tout.min = m;

}

int s = t1.sec + sec;

if(s>xtime)

{

tout.sec = s % xtime;

tout.min += (int)(s/xtime);

}

else

{

tout.sec = s;

}

return tout;

}

};

int main()

{

mytime t1(2,54,79);

cout << "The first time is ";

t1.display();

mytime t2(5,95,45);

cout << "The second time is ";

t2.display();

mytime t3 = t1 + t2;

cout << "The sum is ";

t3.display();

return 0;

}

**OUTPUT:**

The first time is

2 : 55 : 19

The second time is

6 : 35 : 45

The sum is

9 : 31 : 4

1. ***House , Door , Window Problem***

#include<iostream.h>

class door

{

private:

float costPerDoor;

public:

door()

{

}

door(float cost)

{

costPerDoor = cost;

}

float cost(int n)

{

return n \* costPerDoor;

}

};

class window

{

private:

float costPerWindow;

public:

window()

{

}

window(float cost)

{

costPerWindow = cost;

}

float cost(int n)

{

return n \* costPerWindow;

}

};

class house

{

private:

int doorNo, windowNo;

float constArea;

float costPerDoor;

float costPerWindow;

public:

void getData()

{

cout << "Enter the number of doors" << endl;

cin >> doorNo;

cout << "Enter the number of windows" << endl;

cin >> windowNo;

cout << "Enter the area of construction in sq. feet" << endl;

cin >> constArea;

cout << "Enter cost per door" << endl;

cin >> costPerDoor;

cout << "Enter cost per window" << endl;

cin >> costPerWindow;

}

float doorcost()

{

door d(costPerDoor);

return d.cost(doorNo);

}

float windowcost()

{

window w(costPerWindow);

return w.cost(windowNo);

}

};

int main()

{

house myhouse;

myhouse.getData();

cout << "The Total cost for doors is Rs." << myhouse.doorcost() << endl;

cout << "The Total cost for windows is Rs." << myhouse.windowcost() << endl;

return 0;

}

**OUTPUT:**

Enter the number of doors

5

Enter the number of windows

10

Enter the area of construction in sq. feet

1200

Enter cost per door

567.75

Enter cost per window

254.65

The Total cost for doors is Rs.2838.75

The Total cost for windows is Rs.2546.5

1. ***Media , Books , Video Tapes Problem***

#include<iostream.h>

class media

{

protected:

char title[50];

char publ[50];

public:

virtual void read()

{

cout << "Enter the title" << endl;

cin >> title;

cout << "Enter the publication" << endl;

cin >> publ;

}

virtual void show()

{

cout << "The title is " << title << endl;

cout << publ << endl;

}

};

class book:public media

{

private:

int noOfPages;

public:

void read()

{

cout << "Enter the title" << endl;

cin >> title;

cout << "Enter the publication" << endl;

cin >> publ;

cout << "Number of pages " << endl;

cin >> noOfPages;

}

void show()

{

cout << "The title is " << title << endl;

cout << publ << endl;

cout << "Number of pages " << noOfPages << endl;

}

};

class tape:public media

{

private:

int duration;

public:

void read()

{

cout << "Enter the title" << endl;

cin >> title;

cout << "Enter the publication" << endl;

cin >> publ;

cout << "Duration " << endl;

cin >> duration;

}

void show()

{

cout << "The title is " << title << endl;

cout << publ << endl;

cout << "Duration " << duration << endl;

}

};

int main()

{

media \*m;

book b;

tape t;

cout << "------BOOK DETAILS------" << endl;

m = &b;

m->read();

m->show();

cout << "------------------------" << endl;

cout << "------VIDEO TAPE DETAILS--------" << endl;

m = &t;

m->read();

m->show();

cout << "--------------------------------" << endl;

return 0;

}

**OUTPUT:**

------BOOK DETAILS------

Enter the title

Physics

Enter the publication

NewMoon

Number of pages

500

The title is Physics

NewMoon

Number of pages 500

------------------------

------VIDEO TAPE DETAILS--------

Enter the title

Hollywood

Enter the publication

FoxPro

Duration

153

The title is Hollywood

FoxPro

Duration 153

1. ***Student , Streams Class***

#include<iostream.h>

class person

{

public:

char name[50];

char area[50];

};

class examiner

{

public:

int examinerId;

};

class subject

{

public:

int subcode;

int mark;

};

class stream

{

public:

int sid;

char sname[50];

};

class arts:public stream

{

public:

subject algebra,geometry,statistics;

};

class commerce:public stream

{

public:

subject acc,finance,maths;

};

class science:public stream

{

public:

subject phy,chem,maths;

};

class engg:public stream

{

public:

subject ds,oops,dpsd;

};

class student:public person,public engg

{

char college[50];

public:

void getdata()

{

cout << "Enter student name" << endl;

cin >> name;

cout << "Enter area" << endl;

cin >> area;

cout << "Enter stream id" << endl;

cin >> sid;

cout << "Enter stream name" << endl;

cin >> sname;

cout << "Enter subject code for Data Structures , OOPS and DPSD" << endl;

cin >> ds.subcode >> oops.subcode >> dpsd.subcode;

cout << "Enter marks in Data Structures , OOPS and DPSD" << endl;

cin >> ds.mark >> oops.mark >> dpsd.mark;

}

void details()

{

cout << "Student Name : " << name << endl;

cout << "Area : " << area << endl;

cout << "Stream ID : " << sid << endl;

cout << "Stream Name : " << sname << endl;

cout << "Subject Code \t Subject Name \t Mark" << endl;

cout << ds.subcode << "\t Data Structures \t " << ds.mark << endl;

cout << oops.subcode << "\t OOPS \t " << oops.mark << endl;

cout << dpsd.subcode << "\t DPSD \t " << dpsd.mark << endl;

}

};

int main()

{

student s1;

s1.getdata();

s1.details();

return 0;

}

**OUTPUT:**

Enter student name

Srivathsa

Enter area

Triplicane

Enter stream id

3312

Enter stream name

Engineering

Enter subject code for Data Structures , OOPS and DPSD

1842 2283 2314

Enter marks in Data Structures , OOPS and DPSD

95 92 98

Student Name : Srivathsa

Area : Triplicane

Stream ID : 3312

Stream Name : Engineering

Subject Code Subject Name Mark

---------------- ----------------- -------

1842 Data Structures 95

2283 OOPS 92

2314 DPSD 98