OOP in regards to a time class

include "stdafx.h"

#include "time1.h"

#include <iostream>

using namespace std;

time1::time1(int h,int m,int s)

{

setHour(h);

setMinute(m);

setSecond(s);

}

void time1::setHour(int h)

{

hour = (h >= 0 && h < 24) ? h : 0;

}

void time1::setMinute(int m)

{

minute = (m >= 0 && m < 60) ? m : 0;

}

void time1::setSecond(int s)

{

second = (s >= 0 && s < 60) ? s : 0;

}

int time1::getHour()

{

return hour;

}

int time1::getMinute()

{

return minute;

}

int time1 :: getSecond()

{

return second;

}

void time1::printMilitary()

{

cout << (hour < 10 ? "0" : "") << hour << ":" << (minute < 10 ? "0" : "") << minute;

}

void time1::printStandard()

{

cout << ((hour == 0 || hour == 12) ? 12 : hour % 12) << ":" << (minute < 10 ? "0" : "") << minute << ":" << (second < 10 ? "0" : "") << second(hour < 12 ? "AM" : " Pm");

}

time1::~time1()

{

}

#include "stdafx.h"

#include <iostream>

using namespace std;

#include "time1.h"

void incrementSecond(time1 &, const int);

int main()

{

time1 t;

t.setHour(17);

t.setMinute(34);

t.setSecond(25);

cout << "Result of setting all valid values:\n Hour : " << t.getHour() << t.getHour() << "Minute:" << t.getMinute() << "Second:" << t.getSecond();

t.setTime(23, 59, 58);

incrementSecond(t, 2);

return 0;

}

void incrementSecond(time1 &tt, const int count)

{

cout << "Incrementing second" << count << "times : \n Start time:";

tt.printStandard();

for (int i = 0; i < count; i++)

{

tt.setSecond((tt.getSecond()+1)% 60)

if (tt.getSecond() == 0)

{

tt.setMinute((tt.getMinute() + 1) % 60);

if (tt.getMinute() == 0)

{

tt.setHour((tt.getHour() + 1) % 24);

}

}

cout << "\n second + 1";

tt.printStandard();

}

cout << endl;

}

#pragma once

class time1

{

public:

time1(int = 0,int = 0, int = 0);

void setTime(int, int, int);

void setHour(int);

void setMinute(int);

void setSecond(int);

//get functions

int getHour();

int getMinute();

int getSecond();

void printMilitary();

void printStandard();

~time1();

private:

int hour;

int minute;

int second;

};

OOP Class for shapes

#pragma once

class Rectangle1

{

public:

Rectangle1(float l = 1, float w = 1); // constructer

void SetLength(float l);

void SetWidth(float w);

void SetX1(float x1);

float getLength();

float getWidth();

float area();

float permiter();

~Rectangle1(void); //structure remove object when not used = destructer

private:

float length,width,x1,x2,y1,y2; //state

};

#include "stdafx.h"

#include "Rectangle1.h"

//implement

Rectangle1::Rectangle1(float l,float w)

{

SetLength(l);

SetWidth(w);

//width = (w >=1 && w <= 20) ? w:1;

//length = (l >= 1 && l <= 20)? l:1;

}

void Rectangle1::SetLength(float l)

{

length = (1 >= 1 && l <= 20) ? l:1;

}

void Rectangle1::SetWidth(float w)

{

width = (w >= 1 && w <= 20) ? w:1;

}

Rectangle1::~Rectangle1(void)

{

}

float Rectangle1::getLength()

{

return length;

}

float Rectangle1::getWidth()

{

return width;

}

float Rectangle1::area()

{

float area1 = length \* width;

return area1;

}

float Rectangle1::permiter()

{

float per = (length + width) \* 2;

return per;

}

#include "stdafx.h"

#include "Rectangle1.h"

#include <iostream>

using namespace std;

int \_tmain(int argc, \_TCHAR\* argv[])

{

Rectangle1 R1,R2(2),R3(4,8);

//r1 is an object

R1.SetLength(3);

R1.SetWidth(4);

R2.SetWidth(3);

cout << "Result of setting rectangle \n Lenght : " << R1.getLength() << " Width " << R1.getWidth() << endl;

cout << "Result of setting rectangle2 \n Lenght : " << R2.getLength() << " Width " << R2.getWidth() << endl;

cout << "The area of rect1 " << R1.area() << " And perimeter : " << R1.permiter() << endl;

cout << "The area of rect2 " << R2.area() << " And perimeter : " << R2.permiter() << endl;

return 0;

}