

النوت 20 صفحة كمبيوتر
سحب ملونة دينار كويتي واحد
سحب أبيض وأسود ربع دينار فقط

C++ Programming

Week 9

Classes & Object Oriented Programming

(الشرح والأفكار الرئيسية)

نوت السى ++

يحتوي على شرح المواضيع
وأمثلة للأفكار الرئيسية من
هوموركات وامتحانات سابقة.

نوت السى ++

تتكون النوت من عشرة أسابيع.
نوت كل أسبوع يحتوي على الشرح
وتمارين من امتحانات سابقة.

لماذا لا تقتنى الأحدث؟

النوت يتم تنقيحها وتحديثها نهاية
كل أسبوع، راجع eng-hs.net
للتأكد من شرائك الإصدار الأحدث.

(لمن يريد المزيد)

يتوفر على الموقع ملفات الأسابيع
لتمارين وأمثلة إضافية من واقع
امتحانات سابقة.

كثير من الناس عبارة عن
مجرد ظاهرة صوتية.

لاستلام نسخ إلكترونية من نوات الموقع مجاناً (شرح وتمارين محلولة) أو (تمارين وأمثلة إضافية) أو (امتحانات سابقة) على إيميلك قم بزيارة eng-hs.net

لطلب نوات الموقع مطبوعة ملونة مجاناً من تصوير الفرع أمام هندسة أسفل صالون رنيم أو تصوير الجمعية الرئيسية بالسرداب أسفل بيانو قم بزيارة eng-hs.net

النوات متوفرة مجاناً بالموقعين eng-hs.net, eng-hs.com

(1)

م. حمادة شعبان 260 4444 9 info@eng-hs.com

(Introduction to classes)

write a program that reads 3 integers and prints the minimum and the maximum of them.

```
# include <iostream>
using namespace std;
```

```
int main ( )
```

```
{
```

```
int x, y, z, min, max;
```

```
cout << "enter 3 integers: ";
```

```
cin >> x >> y >> z;
```

```
min = x;
```

```
if (y < min)
    min = y;
```

```
if (z < min)
    min = z;
```

```
max = x;
```

```
if (y > max)
    max = y;
```

```
if (z > max)
    max = z;
```

```
cout << "minimum is: " << min << endl;
```

```
cout << "maximum is: " << max << endl;
```

```
return 0;
```

```
}
```

enter 3 integers: **10 3 7**
minimum is: **3**
maximum is: **10**

حل مباشر بدون استخدام
دوال وبدون classes.

عيب هذه الطريقة
صعوبة تقسيم العمل وصعوبة إعادة
استخدام أجزاء منه ببرامج أخرى.

كل الناس يكتمون الأسرار، عدا
أولئك الذين نفضي لهم بأسرارنا.

(Introduction to classes)

write a program that reads 3 integers, and calls functions that return their minimum and maximum, the main function then calls a void function to print the results.

```
# include <iostream>
using namespace std;
```

enter 3 integers: **10 3 7**
minimum is: **3**
maximum is: **10**

```
int minimum (int, int, int);
```

function prototype

```
int maximum (int, int, int);
```

```
void print (int, int);
```

```
int main ( )
```

```
{
```

```
int x, y, z, min, max;
```

```
cout << "enter 3 integers: ";
```

```
cin >> x >> y >> z;
```

function name

```
min = minimum (x, y, z);
```

function call

```
max = maximum (x, y, z);
```

```
print (min, max);
```

```
return 0;
```

```
}
```

حل نفس السؤال السابق
مع استخدام دالة.

عيب هذه الطريقة
تمرير قيم أثناء استدعاء الدوال
يستغرق وقتاً ويتطلب مخازن جديدة.

لا تجعل مساعدتك الآخرين مجرد
فعل ما يتوجب عليهم فعله.

return type

function parameters

int minimum (int a, int b, int c)

function header

```
{
    int min;

    min = a;

    if (b < min)
        min = b;

    if (c < min)
        min = c;

    return min;
}
```

هذه طريقة لحساب
min أو max.

return value

int maximum (int x, int y, int z)

function header

```
{
    if (x > y && x > z)
        return x;

    if (y > z)
        return y;

    return z;
}
```

وهذه طريقة بديلة لحساب
min أو max.

void print (int min, int max)

```
{
    cout << "minimum is: " << min << endl;

    cout << "maximum is: " << max << endl;
}
```

لا توقف طفل عند حدود تعلمك،
فهو مولود في زمن مختلف.

(Converting to classes)

```
# include <iostream>
using namespace std;
class values
{
```

```
private:
```

```
int x, y, z;
```

```
public:
```

```
int minimum ( )
```

```
{
```

```
if (x < y && x < z)
```

```
return x;
```

```
if (y < z)
```

```
return y;
```

```
return z;
```

```
}
```

```
int maximum ( )
```

```
{
```

```
int max = x;
```

```
if (y > max)
```

```
max = y;
```

```
if (z > max)
```

```
max = z;
```

```
return max;
```

```
}
```

```
void print (int min, int max )
```

```
{
```

```
cout << "minimum is: " << min << endl;
```

```
cout << "maximum is: " << max << endl;
```

```
}
```

```
values (int a, int b, int c)
```

```
{
```

```
x = a;
```

```
y = b;
```

```
z = c;
```

```
}
```

```
};
```

مزايا الكلاس

التغلب على عيوب طريقتي البرنامج
الواحد والدوال، ومزايا أخرى كثيرة.

Constructor

لا يتوقف الناس عن اللعب
لأنهم كبروا، بل يكبرون
لأنهم توقفوا عن اللعب.

find the exact output:

```
int main ( )
```

```
{
```

```
    values p1 (10, 3, 7), p2 (8, 6, 9);
```

```
    int min, max;
```

```
    min = p1.minimum ( );
```

```
    max = p1.maximum ( );
```

```
    p1.print (min, max);
```

```
    min = p2.minimum ( );
```

```
    max = p2.maximum ( );
```

```
    p2.print (min, max);
```

```
    return 0;
```

```
}
```

	x	y	z
p1	10	3	7

min	3
	6

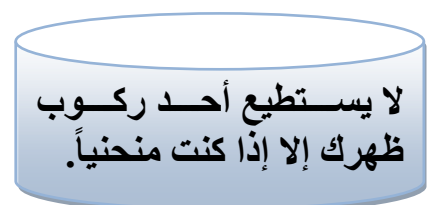
max	10
	9

	x	y	z
p2			

output

```
minimum is: 3
maximum is: 10
```

```
minimum is: 6
maximum is: 9
```



// A simple class to show constructor, default values, set and get functions.

```
#include <iostream>
using namespace std;
```

```
class easy
```

```
{
```

```
    private:
```

```
        int mark;
```

```
    public:
```

```
        void print ( )
```

```
        {
```

```
            cout << "mark is: " << mark << endl;
```

```
        }
```

```
        void set (int x)
```

```
        {
```

```
            mark = x;
```

```
        }
```

```
        int get ( )
```

```
        {
```

```
            return mark;
```

```
        }
```

```
        easy (int x = 0)
```

```
        {
```

```
            mark = x;
```

```
        }
```

```
};
```

Print

* تستخدم لطباعة قيم الـ attributes.
* نوعها void.
* فيها cout.
* لا تستقبل معاملات ().

set

* تستخدم لتبديل قيم الـ attributes.
* تستقبل معاملاً واحداً.
* نوعها void.
* ليس فيها return.

get

* تستخدم لإعادة قيم الـ attributes.
* لها نوع خلاف void.
* لا تستقبل معاملات ().
* تعيد قيمة.

constructor

* يستخدم لإعطاء قيم أو أولية للـ attributes.
* ليس له نوع.
* له نفس اسم الـ class.
* يتم تنفيذه تلقائياً عند تعريف الـ object.
* من نوع الـ class.
* يستخدم لإعطاء قيم أولية.

لا يمكن لأحد أن يشعر بالوضاعة
من دون موافقتك؟

find the exact output:

```
int main ( )
{
```

```
    easy m1 (93), m2;
```

```
    int y;
```

```
    m1.print ( );
```

```
    y = m1.get ( ) + 4;
```

```
    cout << "after curve mark is: " << y << endl;
```

```
    m2.print ( );
```

```
    m2.set (71);
```

```
    m2.print ( );
```

```
    return 0;
```

```
}
```

m1 93 mark

m2 ~~0~~ mark
71

y 97

output

mark is: 93

after curve mark is: 97

mark is: 0

mark is: 71

لا يمكنك أن تبني سمعة
على ما ستفعله لاحقاً.


```
# include <iostream>
using namespace std;
class easy
{
    private:
        int mark;
    public:
        void print ( );

        void set ( int );

        int get ( );

        easy ( int = 0 );
};
```

class definition
غالبا يتم فصله في ملف مستقل عن
class implementation
لثلاثة أسباب رئيسية.

```
void easy::print ( )
{
    cout << "mark is: " << mark << endl;
}
```

```
void easy::set (int x)
{
    mark = x;
}
```

```
int easy::get ( )
{
    return mark;
}
```

```
easy::easy (int x)
{
    set (x);
}
```

main & output
سواء تم كتابة الكلاس كله بملف واحد أو تم فصله في ملفين فإن (main) وكذلك (output) لن يطرأ عليهما أي تغيير.

لا يوجد سبب يحبك الناس من أجله
أفضل من عدم التطلع لما في أيديهم.

find the exact output:

```
int main ( )
```

```
{
```

```
    easy m1 (93);
```

```
    int y;
```

```
    m1.print ( );
```

```
    y = m1.get ( ) + 4;
```

```
    cout << "after curve mark is: " << y << endl;
```

```
    easy m2;
```

```
    m2.print ( );
```

```
    m2.set (71);
```

```
    m2.print ( );
```

```
    return 0;
```

```
}
```

m1 93 ^{mark}

y 97

m2 ~~0~~ ^{mark}
71

output

mark is: 93

after curve mark is: 97

mark is: 0

mark is: 71

لقد لاحظت أن كل من يدعون
للإجهاد قد ولدوا بالفعل.

Find the errors in the following class

```
class Student
```

```
private
```

```
int x;
```

```
public
```

```
int get ( int x )
{
    return x;
}
```

```
int set ( int a )
{
    x = a;
}
```

```
void student ( int a )
{
    x = a;
}
```

```
}
```

مساعدة

الكلاس يحتوي على ثمان
أماكن بها أخطاء، إن طافت منك
واحدة اتصل (9 4444 260).

من النادر أن نفكر فيما نملك، الأغلب
أننا مشغولون بما لا نملك.

```
# include <iostream>
# include <cstring>
using namespace std;
```

```
class Student
{
```

```
    private:
```

```
        int id;
        string name;
        double mark;
```

data members
oR
(attributes)

```
    public:
```

```
        void print ( )
```

```
        {
            cout << "ID = " << id << endl;
            cout << "Name = " << name << endl;
            cout << "Mark = " << mark << endl;
        }
```

```
        void setId (int d)
```

```
        {
            if (d < 0)
            {
                cout << "Invalid ID.\n";
                id = 0;
            }

```

members functions
oR
(methods)

```
            else
                id = d;
        }
```

```
        void setName (string s)
```

```
        {
            name = s;
        }
```

لقد وجدوا أن أفضل طريقة لاستجابة أولادك لنصحك أن تبحث عما يريدون فعله وتنصحهم بأن يفعلوه، أحياناً تفشل.

```

void setMark (double m)
{
    if (m >= 0 && m <= 100)
        mark = m;

    else
    {
        cout << "Invalid mark.\n";
        mark = - 99;
    }
}

```

أحياناً يفضل وضع قيمة غريبة في المتغير للدلالة على حدوث خطأ في عملية الإدخال مثل (-99).

```

int getId ( )
{
    return id;
}

```

```

string getName ( )
{
    return name;
}

```

```

double getMark ( )
{
    return mark;
}

```

```

Student (string s = " ", int d = 0, double m = 0)
{
    setName (s);
    setId (d);
    setMark (m);
}

```

يفضل إعطاء قيم ابتدائية مناسبة للمتغيرات في حال عدم توفر قيمها بالوقت الحالي.

```
};
```

لو لم تكن جزءاً من الحل فعلى الأقل لا تكن جزءاً من المشكلة.

```
int main ( )
```

```
{
```

```
    Student s1, s2;
```

```
    s1.setId (10);
```

```
    s1.setName ("Abdul rahman");
```

```
    s1.setMark (735);
```

```
    s1.print ( );
```

```
    int d;
```

```
    string s;
```

```
    double m;
```

```
    cout << "Enter ID, name & mark for student 2: ";
```

```
    cin >> d >> s >> m;      // -20 Somaiya 81.3
```

```
    s2.setId (d);
```

```
    s2.setName (s);
```

```
    s2.setMark (m);
```

```
    s2.print ( );
```

```
    return 0;
```

```
}
```

	id	name	mark
s1	10	Abdul rahman	-99

d	s	m
-20	Somayia	81.3

	id	name	mark
s2	0	Somayia	81.3

output

Invalid mark.

ID = 10

Name = Abdul rahman

Mark = -99

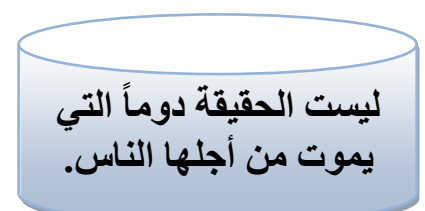
Enter ID, name & mark for student 2: -20 Somaiya 81.3

InvaId ID.

ID = 0

Name = Somaiya

Mark = 81.3



Create a class *Rectangle* that has two attributes *Length* & *Width* of type (integer) and the following member functions:

Rectangle that calls the *setLength* & *setWidth* functions.
(default Length = 1, default Width = 1).

getLength that returns the *Length* value.

getWidth that returns the *Width* value.

setLength that sets the *Length* according to the passed parameter if it is positive, otherwise it shows an error message and sets it to 0.

setWidth that sets the *Width* to the passed parameter if it is positive, otherwise it shows an error message and sets it to 0.

Area that returns the Area of the *rectangle*.

Perimeter that returns the perimeter of the *rectangle*.

```
# include <iostream>
using namespace std;
```

```
class Rectangle
{
private:
```

```
    int Length, Width;
```

```
public:
```

```
    Rectangle (int L = 1, int W = 1)
    {
        setLength (L);
        setWidth (W);
    }
```

ما رأيت ظالماً أشبه
بمظلوم من الحاسد.

```

void setLength (int L)
{
    if (L < 0)
    {
        cout << "Invalid Length\n";
        Length = 0;
    }

    else
        Length = L;
}

```

```

int getLength ( )
{ return Length; }

```

```

void setWidth (int W)
{
    if (W < 0)
    {
        cout << "Invalid Width\n";
        Width = 0;
    }

    else
        Width = W;
}

```

```

int getWidth ( )
{ return Width; }

```

```

int Area ( )
{ return Length * Width; }

```

```

int Perimeter ( )
{ return 2 * (Length + Width); }

```

```
};
```

ما كان للذئب أن يكون ذئباً لو
لم تكن الخرافُ خرافاً.

Classes (class Circle)

Create a class *Circle* that has one attribute *Radius* of type (double) and the following member functions:

Circle that calls the *setRadius* function (default Radius = 0).

getRadius that returns the *Radius* value.

setRadius that sets the *Radius* according to the passed parameter if it is between 0 - 30, otherwise it shows an error message and sets it to 0.

Area that returns the Area of the *circle*.

Perimeter that returns the perimeter of the *circle*.

(The answer is on the next page)



ما يتراءى في ضوء المصابيح
قد يختلف في ضوء الشمس.

```
# include <iostream>
using namespace std;
```

```
class Circle
```

```
{
```

```
private:
```

```
double Radius;
```

```
public:
```

```
Circle (double r = 0)
```

```
{
```

```
    setRadius (r);
```

```
}
```

```
void setRadius (double r)
```

```
{
```

```
    if (r < 0 || r > 30)
```

```
    {
```

```
        cout << "Invalid value\n";
```

```
        Radius = 0;
```

```
    }
```

```
    else
```

```
        Radius = r;
```

```
}
```

```
double getRadius ( )
```

```
{
```

```
    return Radius;
```

```
}
```

```
double Area ( )
```

```
{
```

```
    return 3.14 * Radius * Radius;
```

```
}
```

```
double Perimeter ( )
```

```
{
```

```
    return 2 * 3.14 * Radius;
```

```
}
```

```
};
```

في حساب المساحة والمحيط لابد
نستخدم (Radius) وليس (r).

النوم جيد، والموت أفضل منه،
لكن بالطبع أفضل شيء ألا
تكون قد ولدت من الأساس.

Write a class called Point. The class has a parameterized constructor to initialize both the x-coordinate and y-coordinate of a point object. The class has a member function to display the value of x- and y-coordinates, and another member function to calculate the distance between two points.

```
# include <iostream>
```

```
# include <cmath>
```

```
class Point
```

```
{
```

```
    private:
```

```
        double x, y;
```

```
    public:
```

```
        Point (int a, int b)
```

```
        {
```

```
            x = a;
```

```
            y = b;
```

```
        }
```

```
        void display ( )
```

```
        {
```

```
            cout << " x-coordinate: " << x << endl;
```

```
            cout << " y-coordinate: " << y << endl;
```

```
        }
```

```
        double distance (Point p)
```

```
        {
```

```
            double dx = p.x - x;
```

```
            double dy = p.y - y;
```

```
            double d = sqrt (pow (dx, 2) + pow (dy, 2));
```

```
        }
```

```
};
```

الكثير يفقدون صحتهم أثناء
كسبهم المال ثم ينفقون هذا المال
محاولين استعادة صحتهم.

Create a simple class that is used to represent a Box. each object of this class will store basic information about a box including its height, width, and length. Each object will perform basic calculation associated with boxes as well. This includes volume calculation and surface area calculation. provide a constructor that initializes the class' attributes and a set and get functions for each attribute. Write a program to test your class capabilities.

```
class Box
{
    private:
        double height, width, length;
    public:
        Box (double h, double w, double l)
        { height = h; width = w; length = l; }

        void setHeight (double h)
        {
            if (h < 0)
            {
                height = 0;
                cout << " Invalid height, should be positive \n";
            }
            else
                height = h;
        } // setWidth & setLength are same

        double getHeizht ( )
        { return height; } // sedWidth & getLength are same

        double volume ( )
        { return heigh * Width + length; }

        double surface Area ( )
        { return 2 * (height * width + height * length + width * length; }

};
```

معظم الناس يقضون حياتهم
محاولين إظهار محاسن ليست فيهم.