



المحاضرة الرابعة

البرمجة 1

(*Programming Language 1*)

إعداد

الدكتور المهندس فراس الزين

الكلمات المفتاحية

التنسيق , المخرجات , البرنامج , النظام , نظام تشغيل الأقراص , الألوان , تابع الخرج , تابع الدخل , التعبئة , التقريب , سي ++ , العمليات , العمليات المنطقية , المساواة , الاسناد , صواب , خطأ , التصريح , المدخلات , المعالجة , متغير بإشارة , متغير بلا إشارة .

Format , outputs , program , system , dos , colors , cout , cin , fill , precision , C++ , operations , logic effect , equal , assigned , true , false , declare , inputs , process , signed , unsigned .

تنسيق مخرجات البرنامج Format Outputs

تنسيق الشاشة:

يمكن تحديث المكتبات بدون تضمين مكتبة string ، ومن مميزات تحديث المكتبات الحصول على دوال system التي تمكننا من تنسيق شاشة الإخراج وكذلك استخدام جميع أوامر نظام (DOS).

مثال:

```
1. #include <iostream>
2. using namespace std;
3.
4. main ()
5. {
6.     system("color f0") ;
7.     cout << "new colors";
8.     system("pause");
9. }
```

تضمينه مكتبة `conio.h` حذف اللاحقة `h`.
تحديث المكتبات

تنسيق لونه النص إلى اسود (0) والخلفية إلى أبيض (F)
طباعة نص (سيظهر باللون الأسود)
جعل البرنامج في حالة انتظار

ملاحظات:

1. يتم تمثيل الألوان برقم "سادس عشري" من صفر إلى f حيث يمثل جميع الألوان الأساسية.
2. عند كتابة رقم واحد "color 9" فهذا سيغير لون النص فقط.
3. عند كتابة رقمين "color f0" فإن الأول سيغير لون النص والثاني سيغير لون الخلفية.
4. عند كتابة رقمين متشابهين "color 99" فلن يتغير أي لون، باعتبار أن لون الخط سيثبه لون الخلفية ولذا لن يظهر شيء فذلك يتم تجاهل الألوان وإعادة الألوان الافتراضية.

تنسيق الألوان الافتراضي لمحرر بيئة Microsoft C++:

1. الكلمة المحجوزة تظهر بلون أزرق.
2. الكلمة غير المحجوزة تظهر بلون أسود.
3. التعليقات تظهر بلون أخضر.
4. لغة C++ حساسة لحالة الأحرف (r لا يساوي R)

التنسيق باستخدام (cout):

```
#include <iostream.h>
main(){
    cout << |
    width
    write
    x_delbuf
    x_fill
    x_flags
    x_precision
    x_tie
    x_width
    xalloc
    ~ios
}
```

تحتوي cout على العديد من الخواص الخاصة بالتنسيق ومنها :

- width والتي تعمل إزاحة لليمين من جهة اليسار بمقدار معين
- Fill والتي تقوم بتعبئة الفراغات التي تركتها width برمزم معين.
- Precision والتي تقوم بتقريب الأرقام.

مثال ١:

1. cout.width(20);
2. cout << "Welcome";

النتيجة:

Welcome

مثال ٢:

1. cout.width(20);
2. cout.fill('#');
3. cout << "Welcome";

النتيجة:

#####Welcome

مثال ٣:

الكود:

1. cout.precision(3);
2. cout << 3.449;
3. cout.precision(2);
4. cout << 3.449;
5. cout.precision(3);
6. cout << 3.482;
7. cout.precision(2);
8. cout << 3.482;
9. cout.precision(1);
10. cout << 3.482;
11. cout.precision();
12. cout << 3.482;

النتيجة:

3.45

3.4

3.48

3.5

3

3

العمليات في C++ C++ Operations

العمليات الحسابية:

الرمز	التوضيح
+ - * /	العمليات الرياضية
y / x	القسمة
y \ x	القسمة الصحيحة
y % x	باقي القسمة

مثال :

1. int x = 3 ;
2. int z = 7 ;
- 3.
4. cout << "z + x = " ;
5. cout << z + x ;

النتيجة هو "z + x = "
النتيجة هو 10

ملاحظة : ما داخل الأقواس المفردة أو "المزدوجة" يعتبر نص.

عمليات المقارنة:

الرمز	التوضيح
>	أكبر من
<	أصغر من
> =	أكبر أو يساوي
< =	أصغر أو يساوي
= =	يساوي
! =	لا يساوي

مثال :

1. int x = 3 ;
2. int z = 7 ;
- 3.
4. if (x != z)
5. {
6. cout << "Not equal" ;
7. }

إذا كان x لا يساوي z

فاطببع الجملة "Not equal"
ناتج البرنامج "Not equal" لأنه ناتج الشرط
True "صواب"

العمليات المنطقية Logic Effects:

الرمز	التوضيح
&&	و
	أو
!	نفي

مثال :

```
1. int x = 3 ;  
2. int z = 7 ;  
3.  
4. if ( x > 0 && z > 0 )  
5. {  
6.     cout << "Both numbers positive" ;  
7. }
```

إذا كان x أكبر من الصفر وأيضاً z أكبر من الصفر
(ناتج الشرط True)

ناتج البرنامج "Both numbers positive"

المساواة والإسناد Equal and Assigned

الإسناد: هو إعطاء المتغير قيمة:

مثال:

```
1 int x ;  
2 x = 7 ;  
  
3 cout << x;
```

الناتج 7

المساواة: هو مقارنة قيمتين:

مثال:

```
1. int x = 7 ;  
2. int y = 7 ;  
3.  
4. cout << x == y;  
5. cout << x > 3  
6.  
7. int z = x == 8  
8. cout << z ;
```

الناتج صواب (True) "سيطبع 1 في الشاشة"
الناتج خطأ (False) "سيطبع 0 في الشاشة"

اسناد ناتج المقارنة للمتغير z
الناتج False لأنه x يساوي 7 وليس 8
وستتم طباعة 0 على الشاشة.

ملاحظة:

$x = y$ تعني (هل أن x يساوي y) وهي عملية مقارنة ناتجها إما صواب أو خطأ.

تحويل المعادلات الرياضية إلى معادلات برمجية:

	المعادلة الرياضية	المعادلة البرمجية
1)	$z = x^2 + x + 7$	$Z = \text{pow}(x, 2) + x + 7 ;$
2)	$z = \frac{x+1}{y+1}$	$Z = (x+1) / (y+1) ;$
3)	$z = \frac{(x^2 + x + 7)^2}{y + x + 1}$	$Z = \text{pow}(\text{pow}(x, 2) + x + 7, 2) / (y + x + 1)$

توجد الدالة pow ضمن المكتبة math.h لذلك يجب تضمين المكتبة math.h في البرنامج، المزيد من الدوال الرياضية في الفصل الخامس .

س: كيف تكتب المعادلة التالية برمجياً؟

$$y = \begin{cases} x+1 : x < 0 \\ x^2 + x + 7 : x > 0 \end{cases}$$

1. if (x < 0)
2. {
3. Y = x + 1 ;
4. }
5. else
6. {
7. Y = pow(x , 2) + x + 1 ;
8. }

ج: إذا كان x أصغر من الصفر

وإذا كان غير ذلك (جميع الحالات التي لا توجد في الشرط السابق مثل $x > 0$ أو $x = 0$)

س: كيف تكتب المعادلة التالية برمجياً؟

$$R = x^{y^2}$$

$$R = \text{pow}(x, \text{pow}(y, 2)) ;$$

ج:

س: كيف تكتب المعادلة التالية برمجياً؟

$$Y = \sqrt{3^2}$$

$$Y = \text{sqrt}(\text{pow}(3, 2)) ;$$

ج:

أولوية العمليات الحسابية:

- ٦) ما بداخل الأقواس.
- ١- الأس.
- ٢- الضرب ثم القسمة.
- ٣- الجمع أو الطرح.

برنامج لإيجاد مساحة المستطيل:

من المهم :

- فهم فكرة البرنامج.
- تحويل الفكرة إلى خطوات منطقية، وأهمها معرفة معادلة مساحة المستطيل.

$$\text{مساحة المستطيل} = \text{الطول} \times \text{العرض}$$

إذاً نحتاج إلى ٣ متغيرات ، متغيرين سيقوم المستخدم بإدخالهما (الطول والعرض) ومتغير سيحتوي على ناتج ضرب المتغيرين (المساحة).

خطوات الحل البرمجي:

(١) التصريحات Declare	(تعريف المتغيرات).
(٢) المدخلات Input	(إسناد قيم للمتغيرات "الطول والعرض").
(٣) المعالجة Process	(الضرب).
(٤) المخرجات Output	(مساحة المستطيل).

1. int height, width, area;
2. cin >> height >> width;
3. area = height * width
4. cout << area;
- 5.

تعريف ثلاثة متغيرات مع نفس النوع في سطر واحد :
استقبال قيمته مع المستخدم:
تخزينه نتيجة ضرب القيمته في متغير المساحة
عرض الناتج (مساحة المستطيل)
الناتج:

9

3

27Press any key to continue_

كما نلاحظ أن الناتج في المثال السابق غير واضح وعند تنفيذ البرنامج تظهر شاشة سوداء لا يوجد فيها أي معلومات تساعد مستخدم البرنامج في معرفة ما المطلوب منه وماذا يجب أن يعمل..

تنسيق المخرجات:

أعادة المثال السابق:

```
1. int height, width, area;
2. cout << "Enter Height: ";
3. cin >> height;
4. cout << "Enter Width: ";
5. cin >> width;
6. area = height * width;
7. cout << "-----\n";
8. cout << "Result is: " << area << endl;
```

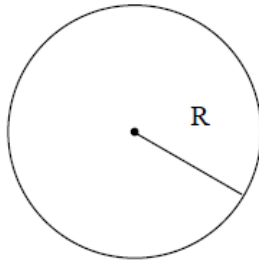
طباعة نص يطلب إدخال الطول
استقبال الطول من المستخدم
طباعة نص يطلب إدخال العرض
استقبال العرض من المستخدم

طباعة خط أفقي وسطر جديد
طباعة " النتيجة هي " ثم ناتج الضرب ثم سطر جديد
الناتج:

```
Enter Height: 7
Enter Width: 8
-----
Result is: 56
Press any key to continue_
```

تمرين :

أكتب برنامج لحساب مساحة الدائرة إذا علمت أن:



$$\text{Circle} = \pi * R^2$$

حيث أن :

R : نصف القطر (معطى).

π : 3.14 (ثابت).

الفرق بين signed و unsigned:

signed جعل نوع البيانات يقبل القيم السالبة وهو الافتراضي حتى إذا لم يكتب، بينما unsigned لا يقبل القيم السالبة، حيث يرفض أي قيمة سالبة وتظل قيمته عشوائية إلى أن يتم إسناد قيمة موجبه إليه.

مثال:

1. #include <iostream.h>	
2.	
3. main ()	
4. {	
5. signed int x;	
6. unsigned int z;	
7. x = 100;	
8. z = 100;	
9. cout << x << endl;	100
10. cout << y << endl;	100
11. x = -100;	
12. z = -100;	
13. cout << x << endl;	-100
14. cout << y << endl;	4294967196
15. }	

الناج

والرقم 4294967196 عبارة عن رقم عشوائي جاء من الذاكرة نتيجة لأن المتغير y لم يسند له أي قيمة، وذلك لأن النوع unsigned يجعل المتغير يرفض أن يسند إليه قيمة سالبة.