

الفصل الثاني جمل الإدخال والإخراج INPUT / OUTPUT STATEMENTS

1.5 مقدمة

يتناول هذا الفصل أوامر إدخال القيم الحسابية والرمزية وأوامر إخراج وطباعة نتائج البرامج والتعبير الحسابية والرمزية وطباعة المعطيات المختلفة حسب الحاجة. توفر لغة C++ مجموعة من الدوال والأوامر للإدخال والإخراج سنتطرق لها بالفقرات القادمة.

2.5 أوامر الإدخال والإخراج Input/Output Orders

كل برنامج يجب أن تكون له مدخلات تتمثل بالبيانات التي تعطي للحاسب وكذلك له مخرجات تبين النتائج التي تم الحصول عليها من البرنامج. أن أوامر الإدخال والإخراج في لغة C++ تتوفر ضمن الدالة (iostream.h) وهي موجودة ضمن المكتبة القياسية للغة C++، ويجب ذكر أسم هذه الدالة محاطاً بالعلامتين < > وفي أول سطر بالبرنامج بعد كلمة (#include) وكما يأتي :

```
#include <iostream.h>
```

حيث : io : يمثل مختصر لـ input/output أي الإدخال والإخراج

stream : مكتبة قياسية خاصة بالإدخال والإخراج

h : Header file أي ملف الدليل

1.2.5 أمر الإدخال (cin)

يستخدم الأمر (cin) مع العلامة (>>) لإدخال (قراءة) المتغيرات الحسابية أو الرمزية للبرنامج وهي تعني (أقرأ القيمة المطبوعة على شاشة التنفيذ وأخزنها في موقع الذاكرة الذي يشار إليه بواسطة المتغير الموجود بعد العلامة (>>)) وفي حالة وجود أكثر من متغير يراد قرأته يمكن كتابة صيغة الإدخال بالشكل التالي:

```
cin>> var1>>var2>>var3>>.....
```

مثال: أكتب جملة قراءة مناسبة لقراءة للمتغيرات التالية :

```
a=12.6      b=52.3      c=1200
```

```
cin>>a>>b>>c;
```

الجواب :

حيث يتم إدخال قيم المتغيرات في شاشة تنفيذ

2.2.5 أمر الإخراج (cout)

الإخراج (وهو طباعة النتائج على شاشة الحاسب) فيستخدم الأمر (cout) مع العلامة (<<) أي أطبع قيمة المتغير الذي يأتي بعد العلامة (<<) وفي حال وجود أكثر من متغير يراد طباعته يمكن كتابة صيغة الإخراج بهذه الطريقة:

```
cout<<var1<<var2<<var3.....
```

تتعامل صيغة الإخراج مع نوعين من المخرجات حسب ما توضع بعد علامة (<<) :

1) طباعة النصوص الحرفية أو الرمزية : لطباعة نص حرفي أو سلسلة حرفية (string) يجب ان يحاط النص الحرفي بين علامتي اقتباس مزدوجة " " فأن ما موجود بين علامتي الاقتباس يتم طباعته دون أدنى تغيير. مثال على ذلك:

```
cout<<"my nams is Zeki";
```

```
cout<<"5+4=9";
```

2) أما إذا كان ما موجود بعد العلامة (<<) ليس محدد بين علامتي اقتباس فعند ذلك سيعامل ما موجود بعدها على أنه معرف والمعرفات هنا تكون على واحدة من الحالات أدناه :

i. أما أن تكون مقادير ثابتة (قيم حسابية) مثل القيم (78.456،-123،4567...الخ) فهي تطبع مباشرة على الشاشة دون تغيير، مثلا:

```
cout<<3456;
```

هنا سيتم طباعة العدد (3456) على الشاشة

ii. أو تكون على شكل تعبير حسابي (expression) أي مقادير تفصل بينها العوامل الرياضية مثل (*،-،+،/،...) وبهذه الحالة فسيتم استخراج قيمة العملية الحسابية و طباعتها على الشاشة، مثال على ذلك :

```
cout << 34 + 56 ;
```

هنا سيتم طباعة نتيجة العملية وهي (90).

iii. تستخدم لطباعة متغير ما سواء كان صحيح أو حقيقي أو رمزي حسب نوعه وهنا يجب أولاً أن يعرف نوع المتغير (int، float، char، double...الخ) وبعد ذلك تعطى قيمته للبرنامج. مثال على ذلك:

```
cout<<x;
```

فتطبع قيمة (x) وفقاً لما تم تعريفه في البرنامج وحسب نوعها كما ذكر

3.2.5 رموز الدلالة Directing Characters

وهي حروف خاصة عادة تستخدم مع الشرطة العكسية (\) لإعطاء تأثير معين يلاحظ ضمن مخرجات البرنامج لذا فهي تستخدم مع أمر الإخراج (cout). وهذه تسمى أيضا سلاسل الهروب (Escape Sequences) فالشرطة المعكوسة (\) التي تسبق بعض الأحرف تخبر المترجم بان هذا الحرف الذي يلي الشرطة المعكوسة ليس له نفس المعنى كما لو ظهر الحرف بنفسه دون هذه الشرطة المعكوسة (\) هذه السلاسل يتم كتابتها كرمزين دون وجود فراغ بينهما. الجدول أدناه يوضح أنواع هذه الرموز والتأثير الذي تحدثه ومما ينبغي الأشاره إليه أن هذه الرموز يجب أن تكتب بين حاصرتين مزدوجتين " " .

الرمز	النتائج (التأثير على المخرجات)
\a	صوت أو صفير (Beep)
\b	الإرجاع خانه واحدة للخلف (Backspace)
\f	التغذية (feed)
\n	سطر جديد (new line)
\r	الإعادة أو الرجوع (carriage return)
\t	الإزاحة الأفقية (horizontal tabulator)
\v	الإزاحة العمودية (vertical tabulator)
\\	الشرطة المعكوسة (Backslash)
\'	حاصرة مفردة (single quota)
\"	حاصرة مزدوجة (double quota)

مثال :

```
cout<<"First line\nSecond line";
```

نتيجة الطباعة لهذا المثال ستكون في سطرين متتاليين على النحو التالي :

First line

Second line

4.2.5 بعض الصيغ المهمة في عمليات الإدخال والإخراج

Formatted Consol for I/O Operations

تدعم ++C عدد من الصفات التي من الممكن ان تستخدم لصياغة او تنظيم طريقة ظهور المخرجات والموضحة بالجدول التالي، هذه الدوال تستخدم مع الموجة (iostream) أو ما يكافئها مع الموجه (iomanip) وهي تستخدم بالترافق مع الأمر (cout) ، والصيغة العامة لها هي:

```
cout.function
```

لاحظ هنا تم استخدام النقطة (.) بدلا من (<<) (>>)

وظيفة الدالة	دوال مع الموجه #include<iomanip >	دوال مع الموجه #include<iostream>
تحدد حجم الحقل المطلوب لعرض قيم المخرجات	setw()	width ()
تحدد عدد المراتب بعد الفارزة عند عرض القيم الحقيقية	setprecision ()	precision ()
تحدد نوع الرمز الذي سيستخدم لملئ الجزء غير المستخدم في الحقل المحدد لعرض قيمة معينة	setfill()	fill ()
تحدد إشارة للمسيطر لتحديد نوع الصياغة المطلوبة (مثل طباعة القيمة من اليمين أو اليسار، ملء السطور)	setiosflags ()	setf ()
تستخدم لإلغاء الصياغة المحددة بالإيعاز السابق	unsetiosflags ()	unsetf ()

مثال :

```
cout.width(5);
```

```
cout<<354;
```

المخرجات ستكون كالتالي :

		3	5	4
--	--	---	---	---

أي إن المترجم سيحدد خمس مواقع لطباعة القيمة، ويبدأ الطباعة من اليمين، لذلك سيكون هنا فراغين في اليسار.

ملاحظة://:

1) تأثير الدالة () width يستمر لأمر طباعة واحد فقط، فإذا كان هنا أكثر من أمر طباعة فنستخدم width() مع كل أمر طباعة.

2) يستخدم الأمر (fill) لمأ الفراغات، ويجب أن توضع بين قوسي الأمر (fill) الرمز المطلوب طباعته (بما انه رمز فيجرب أن يحدد بحاصرات مفردة) أما إذا لم يحدد ماهية الرمز المطلوب طباعته في الحقول الفارغة (عند تحديد حجم الحقل لطباعة قيمة معينة) فإن المترجم سيتركها فارغة كما في المثال السابق.
مثال :

```
cout.fill ( '* ' );
cout.width ( 7 );
cout << 345 ;
```

في هذه الحالة فان الحقول الفارغة ستمأ بالعلامة (*) وستكون النتيجة:

*	*	*	*	3	4	5
---	---	---	---	---	---	---

3) الأمر (precision) يستخدم عادة مع الأعداد الحقيقية لتحديد عدد المراتب بعد الفارزه كما بالأمثلة أدناه:

```
cout.precision ( 9 );
cout<< 354.045898;
```

ستكون نتيجة الطباعة للمثال أعلاه هي العدد 354.045898

```
cout.precision(6);
cout<<354.045898;
```

ستكون نتيجة الطباعة هنا هي العدد 354.046

```
cout.precision(3);
cout<<354.045898;
```

ستكون نتيجة الطباعة هنا هي العدد 354

```
cout.precision(2);
cout<<354.045898;
```

ستكون نتيجة الطباعة هنا هي العدد 3.5e+002

*يجب الانتباه هنا انه إذا لم يحدد عدد المراتب للأعداد الحقيقية فإن المترجم يفرضها ستة مراتب

4) الدالة (setf) من الممكن أن تأخذ معامل واحد أو معاملين (وسيط أو اثنين)، ووفقا لهذه المعاملات سيحدد واجبها وكما يأتي:

i) استخدام وسيط واحد مع الدالة (setf) والصيغة العامة لها هي:

cout.setf (arg) ;

الجدول أدناه يبين وظيفة الدالة (setf) عند استخدامها وسيط واحد ووفقا لقيمة الوسيط المقابل لها :

وظيفته	قيمة المعامل (flag)
تستخدم base indicator في المخرجات	ios::showbase
تطبع العلامة الموجبة (+) قبل الأرقام الموجبة	ios::showpos
تظهر الفارزة والأصفار	ios::showpoint
حذف الفراغات (white space) في المخرجات	ios::skipws
تدفق الناتج بعد كل عملية إدخال	ios::unitbuf
كتابة القيم الكسرية كقيم كسرية ثابتة	ios:: fixed
كتابة القيم الكسرية بالعلامة العلمية	ios::scientific

(ii) الدالة مع وسيطين وتكون الصيغة العامة لها كما يأتي:

cout.setf(arg1,arg2);

وظيفة الدالة	قيمة المعامل (arg1)	قيمة المعامل (arg2)
ملء السطور من اليسار	ios::left	ios::adjustfield
ملء السطور من اليمين	ios::right	ios::adjustfield
إظهار العلامات الرياضية (الإشارة الموجبة والسالبة)	ios::internal	ios::adjustfield
العلامة العلمية	ios::scientific	ios::floatfield
علامة النقطة الثابتة	ios::fixed	ios::floatfield

مثال : أعط مخرجات جزء البرنامج التالي:

cout.fill ('@') ;

cout.precision (3) ;

cout.setf(ios::internal,ios::adjustfield) ;

cout.setf (ios:: scientific,ios::floatfield);

cout.width (15) ;

cout << -12.34567 <<"\n" ;

الحل:

-	@	@	@	@	1	.	2	3	5	e	+	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

توضيح : تلاحظ أن الإيعاز الأول هو لملئ الفراغات بالرمز (@) ، إما الإيعاز في السطر الثاني فهو يمثل عدد المراتب بعد الفارزة للرقم الحقيقي وهي هنا (3) ، الإيعاز الثالث فهو يستخدم معاملين أو وسيطين لإظهار العلامة الرياضية، الإيعاز الرابع في السطر الرابع يستخدم لإظهار العلامة العلمية، ثم يليه تحديد عدد المواقع التي ستطبع بها القيمة وهي هنا (15 موقع) وأخيرا تم إدخال القيمة المراد طباعتها كما بالنتيجة أعلاه.

مثال : أعط مخرجات البرنامج التالي:

```
//modify precision
#include <iostream.h>

main ( )
{
    double f = 3.14159;
    cout.unsetf (ios::floatfield );           // floatfield not set
    cout.precision(5);
    cout << f << "\n";
    cout.precision(10);
    cout << f << "\n";
    cout.setf( ios::fixed, ios::floatfield ); // floatfield set to fixed
    cout << f << "\n";
    return 0;
}
```

مخرجات البرنامج تكون كالتالي:

```
3.1416
3.14159
3.1415900000
```

مثال : لو كانت قيمة كل من

```
x=12.6  y=52.3  m=-100  n=1200  s=red  t=blue
```

بين كيف ستكون نتيجة جمل الطباعة التالية:

```
cout<<"x="<<x<<" y="<<y<<" m="<<m<<"\n';
```

```
cout<<"n="<<n<<" s="<<s<<" t="<<t;
```

الجواب : تكون نتيجة الطباعة كالتالي على سطرين وكالتالي:

```
x=12.6 y=52.3 m=-100
```

```
n=1200 s=red t=blue
```

مثال: بين مخرجات البرنامج التالي:

```
#include <iostream.h>
```

```
main()
```

```
{  
    cout<<"matrix";  
    cout<<"matrix\n";  
    cout<<"matrix\n\n";  
    cout<<"matrix\n\n\n";  
    cout<<"matrix";  
    return 0;  
}
```

الجواب:

السطر الأول : matrixmatrix

السطر الثاني : matrix

السطر الثالث: فارغ

السطر الرابع: matrix

السطر الخامس: سطر فارغ

السطر السادس: سطر فارغ

السطر السابع: matrix

مثال: أكتب برنامج لحساب مساحة ومحيط دائرة.

الحل:

```
# include <iostream.h>
#include <math.h>
main()
{
const double pi=3.14159;
double radius,area,circum;
cout<<"Enter Radius\n";
cin>>radius;
area=pi*pow(radius,2);
circum=2*pi*radius;
cout<<"The value of circum="<<circum<<"\n";
cout<<"The value of area="<<area;
return 0;
}
```

مثال: اكتب برنامج لإيجاد قيمة العلاقة التالية:

$$y=3x^2-2x+4$$

الحل:

```
#include <iostream.h>
main( )
{
float x,y;
cin>>x;
y=3*x*x-2*x+4;
cout<<"y="<<y;<<endl;
```

```
return 0;
}
```

مثال: أكتب برنامج لتحويل درجة حرارة مقاسه بالفهرنهايت إلى درجة مئوية.

```
#include <iostream.h>
main( )
{
float f;
cout<<"Enter temperature degree in Fahrenheit\n";
cin>>f;
float c=(5*(f+32))/9;
cout<<c<<endl;
return 0;
}
```

مثال : أكتب برنامج لحساب معدل طالب في المواد التالية :

الوحدة	الدرجة	المادة
7	70	رياضيات
8	66	ميكانيك
5	75	برمجة
5	71	كهربائية
3	82	رسم هندسي

الحل :

```
#include <iostream.h>
main( )
{
int Ma,Me,Pr,El,Dr;
double Ume,Upr,Uel,Udr,Uma,Usum;
```

```
double Mean,Sum;
Ma=70;
Me=66;
Pr=75;
El=71;
Dr=82;
Uma=7;
Ume=8;
Upr=5;
Uel=5;
Udr=3;
Sum= Ma * Uma + Me * Ume + Pr * Upr + El * Uel + Dr * Udr;
Usum= Uma + Ume + Upr + Uel + Udr;
Mean= Sum / Usum;
cout<<"Mean="<<Mean<<endl;
return 0;
}
```

أسئلة حول ملفات الإدخال والإخراج

س1/ اكتب برنامج بلغة C++ لتحويل (42200 s) إلى ما يقابلها بالساعات والدقائق والثواني؟

س2/ اكتب برنامج لغة C++ لإيجاد حاصل ضرب ومعدل ثلاث أعداد؟

س3/ اكتب برنامج بلغة C++ لحساب عدد السكان P باستخدام معادلة نمو السكان التالية :

$$P = \frac{700P_0}{P_0 + (700 - P_0)e^{-(T/10)}}$$

حيث P_0 تمثل عدد السكان الحالي ويساوي 100000

T تمثل الزمن ويساوي هنا خمس سنوات

س4/ اكتب برنامج بلغة C++ تحسب فيه قيمة الضغط P من معادلة الغازات التالية:

$$P = \frac{RT}{V} + \frac{A}{V^2} + \frac{B}{V^3} + \frac{C}{V^4}$$

حيث (C,B,A,T,R) ثوابت عددية , علما بان قيمة الحجم V تساوي 1.

س5/ مثلث قائم الزاوية , طول كل من ضلعيه 3.7 , 4.5 على التوالي, اكتب برنامج يقرأ فيه أطوال

الأضلاع المعلومة, ويحسب طول الوتر, ومساحة المثلث, والزاويتين المتقابلتين للضلعين, ثم أطلع النتائج

بحيث تطبع لكل نتيجة العنوان المناسب.

س6/ صحح جزء البرنامج التالي

```
#include<iostream>
Main() {
Char gap = ' ';
Int m ; n ;
float a, b;
char c1, c2
int (a + m) = 12 ;
cin >> a >> b >> m >> n ;;
cout << a+b << c1 ;
gap = a + c2 ;
m = a / b ;
cin > c2 ;
cout << n = a * b ;
```

```
if ( a = b) cout << " equal" ;  
else cout << a not equal b ;
```