قسم الرياضيات والحاسوب / مرحلة اولى

برمجة مهيكلة Structured Programming

المحاضرة (3)

الهدف من المحاضرة : هو التعرف على ادوات الهروب ، المتغيرات والثوابت ، المعاملات الخاصة بتغيير قيمة متغير، الثوابت وذلك من خلال تطبيق التمارين ، اضافة سيتم التعرف على دالة if الشرطية.

أدوات الهروب:

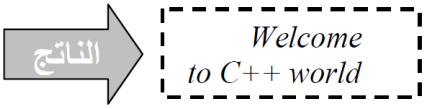
هي أدوات تستخدم للتحكم في موقع المخرجات من حيث التحرك الأفقي أو الرأسي عبر السطور حيث تستخدم \n للنزول سطراً، و \t للتحرك مسافة أفقية.

أدوات التعليق:

يمكن التعليق على البرنامج لتوضيح عمل دالة أو أمر معين لمن يقرأ البرنامج دون أن يؤثر ذلك على عمل البرنامج حيث لا يعتبر التعليق جزأً من البرنامج، تستخدم الأداة // لإضافة تعليق في سطر واحد، بينما يوضع التعليق متعدد الأسطر بين العلامتين */التعليق/* .

يمكن إعادة كتابة البرنامج السابق مع إضافة أدوات الهروب والتعليق:

```
#include<iostream.h>
void main()
{ //start of main function
 cout<<"\twelcome\n to C++ world";
}  /*end of main function
      this a multi_line comment*/
```



تمرين 1

1. أنشيء مخططاً انسيابياً لبرنامج يحسب مساحة ومحيط المستطيل عند إدخال الطول و العرض؟
2. أكتب برنامج بلغة C++ يظهر على الشاشة النص " This is my first program in C++ language"

المتغيرات والثوابت Variables & Constants

المتغير هو موقع في الذاكرة يمكن تخزين القيم واسترجاعها عند الحاجة ، سعة الذاكرة التي تخصص للمتغير تعتمد على نوعه ، والذي يمكن ان يكون عددا صحيحا أو حقيقيا أو متغيرا حرفيا ، الجدول التالي يوضح أنواع المتغيرات في لغة C++:

النوع	الحجم	التمثيل	القيم
int	2بايت عدا في ويندوز 95 أو ويندوز NT فحجمه 4 بايت	الأعداد الصحيحة	-32768 إلى32767 إلا في حالة 4 بايت فالقيم من -2147483648إلى2147483647
short int	2 بايت	الأعداد الصحيحة	-32768 إلى32767
long int	4 بايت	الأعداد الصحيحة	-2147483648 إلى2147483647
float	4 بايت	الأعداد الحقيقية	-1.2e-38 إلى3.4e38
double	8 بايت	الأعداد الحقيقية	-2.2e-308 إلى 1.8e308
char	1 بايت	الأحرف	256 حرفا " أحرف آسكي "

ملاحظات:

- يستخدم التعبير signed و unsigned للمتغيرات الرقمية لتحديد هل تشمل الأرقام الموجبة والسالبة أم لا، فالتعبيرunsigned يدل على الأعداد الموجبة فقط مما يعني تخصيص مدى القيم في الأرقام الموجبة

فقط، أما signed فتدل على الأعداد الموجبة والسالبة وإذا لم يذكر أي من التعبيرين فيؤخذ على انه signed.

مثلا التعبير **unsigned short int** يمثل من **0 الى 65535**.

أما التعبير signed short int أو short int فيمثل الأعداد من 32768- الى 32767.

- الأنواع int, short int, long int جميعها تمثل الأعداد الصحيحة والإختلاف في المدى الذي تمثله.
- الأنواع float, double جميعها تمثل الأعداد الحقيقية والإختلاف في المدى الذي تمثله.
- Char تمثل كل أحرف آسكي وهي تشمل الحروف اللغوية والعلامات والأقواس والأرقام غير انها يتعامل معها على انها أحرف أي لا تخضع للعمليات المنطقية والحسابية.

تعريف المتغيرات:

يتم تعريف المتغير بذكر نوعه ثم اسمه بحيث يفصل بينهما فراغ وكأي سطر برمجي لابد من الانتهاء بفاصلة منقوطة الصيغة العامة لتعريف المتغيرات

variable_type variable_name ;

مثلاً التعريف;int x يخص تعريف متغير إسمه x على أنه عدد صحيح، والتعريف;float area لعدد حقيقي area . يمكن تعريف عدة متغيرات في سطر واحد إذا كانت من نوع واحد فالتعريف ;char a,b,c يعني أن a,b,c متغيرات حرفية.

قيود على إسم المتغير

- ❖ يجب ألا يحتوي الإسم على فراغ.
- ❖ يجب ألا يحتوي الإسم على العلامات الخاصه مثل +،-،*،/.،#،@..
- ❖ يجب ألا يبتدئ الإسم برقم.

❖ يجب ألا يمثل الإسم كلمة من كلمات اللغة المحجوزة وهي الكلمات المستخدمة في الأوامر مثل if, else, int, float, long.

ملاحظات:

- يمكن أن يكون الرقم في وسط أو آخر الإسم.
- الكلمات المحجوزة تظهر بتنسيق مميز في معظم المصرفات التي تكتب فيها البرامج.
- هنالك حساسية لحالة الحرف في تسمية المتغير أي أن area مثلاً تختلف عن AREA .

تعيين قيم المتغيرات:

تتم نسبة القيم إلى المتغيرات بإستخدام معامل الإسناد = والذي ينسب القيمة التي على يمينه للمتغير على يساره، كما في المثال التالي:

x=5; حيث تنسب القيمة 5 للمتغير x وهذا يعني تخزين هذه القيمة للموقع المحجوز لـ x .

المعاملات الرياضية Mathematical Operators

لكتابة القوانين والمعادلات الخاصة بنظام معين تستخدم هذه المعاملات الرياضية ، وهي موضحة في الجدول التالي:

المعامل	العملية الرياضية
+	الجمع
–	الطرح
*	الضرب
/	القسمة
%	باقي القسمة الصحيح مثلا 23%7=2

كل هذه الأدوات تعمل في الاتجاه من اليسار إلى اليمين

الأولوية في تنفيذ العمليات الرياضية

تجعل لغة C++ الأولوية في التنفيذ لعمليات الضرب والقسمة الصحيح على السواء ثم عمليتي الجمع والطرح. وإذا احتوى تعبير رياضي على أكثر من عملية لها نفس الأولوية فإن العملية التي على اليسار تنفذ أولاً. وفي جميع الأحوال فإن العمليات داخل الأقواس تنفذ أولاً.

مثال: ما قيمة x في كل مما يأتي:

- X=4+2*5
- X=5%3*7+2
- X=5+(4%3)

الحل:

- X=4+10=14
- X=2*7+2=14+2=16
- X=5+1=6

ملاحظة:

يمكن استخدام معامل الإسناد= مع أي من المعاملات الرياضية عندما يراد إسناد قيمه لمتغير معين بدلالة نفس المتغير كما هو مبين في الجدول:

التعبيـــــر	التعبير المختصر
x=x+3	x+=3
a=a*2	a*=2

مثال: اكتب برنامج بلغة C++ يحسب مساحة ومحيط المستطيل عند ادخال الطول والعرض؟

الحل: في هذا البرنامج نتعامل مع اربعة متغيرات هي الطول length، العرض width المساحة area، والمحيط perimeter. هذه المتغيرات لا يشترط ان تكون أعداداً صحيحة int بل يمكن ان تحتوي على كسور

لذا فأن من المناسب لها ان تعرف اعداداً حقيقية float، والطول والعرض سنحصل على قيمها عن طريق إدخال المستخدم. أما المساحة والمحيط فتحسبان بالعلاقتين المعروفتين:

المساحة= الطول * العرض

المحيط= 2*(الطول + العرض)

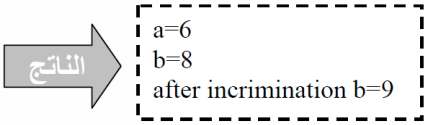
لذا يكون البرنامج كالآتي:

```
#include<iostream.h>
void main()
{
float length,width,area,perimeter;   //variables definition
cout<<"\n\n\tCalculation of the Area & perimeter for the Rectangle  ";
cout<<"\n\tenter the length of the rectangle ";
cin>>length;
cout<<"\n\tenter the width of the rectangle ";
cin>>width;
area=length*width;
perimeter=2*(length+width);
cout<<"\n\tthe area of this rectangle is = "<<area;
cout<<"\n\tthe perimeter of this rectangle is = "<<perimeter;
}
```

المعاملات الخاصة بتغيير قيمة متغير

- معامل الزيادة بواحد "++" ويمكن ان يكون قبلي pre_incrimination بحيث تتم الزيادة قبل تنفيذ الأمر الحالي ويكتب المعامل قبل اسم المتغير او بعده post_incrimination بحيث تتم الزيادة بعد تنفيذ الأمر الحالي مباشرة ويكتب المعامل بعد اسم المتغير.
- معامل النقصان بواحد "--" ويمكن ان يكون قبلي pre_incrimination بحيث يكون النقصان قبل تنفيذ الأمر الحالي ويكتب المعامل قبل اسم المتغير او بعده post_incrimination بحيث يتم النقصان بعد تنفيذ الأمر الحالي مباشرة ويكتب المعامل بعد اسم المتغير.

```
#include<iostream.h>
void main()
{
int a=5,b=8;
cout<<"\na= "<<++a;
cout<<"\nb= "<<b++;
cout<<"\n after incrementation b= "<<b;
}
```



الثوابـــــــــــت Constants

الثابت يأخذ نفس الحيز التخزيني للمتغير غير ان قيمته غير قابلة للتغير. يتم الاعلان عن المتغير باستخدام الكلمة المحجوزة const يعقبها نوع الثابت ثم اسناد قيمة لهذا الثابت كما في البرنامج التالي الذي يحسب مساحة الدائرة عند إدخال قيمة نصف قطرها.

```
#include<iostream.h>
void main()
{
float radius,area;
const float pi;
cout<<"enter the circle radius:";
cin>>radius;
  area=pi*radius*radius;
  cout<<"\nthe area of a circle is = "<<area;
  }
```

ملاحظات

- ✓ لابد من اسناد قيمة للثابت عند الإعلان عنه.
- ✓ إذا لم يحدد نوع الثابت فإنه يعتبر افتراضيا عدد صحيح int.

تمرين:

1- إذا كان y متغيرا من نوع int فما قيمته في كل مما يأتي:

1. y=10*7%(5+8-3)
2. y=18/5+4

2- أكتب برنامجاً لحساب الدخل الإجمالي total_income لموظف إذا علم الراتب الأساسي bs، بدل النقل t " 5% من الراتب الأساسي" ، بدل خبره e حيث يحسب الدخل الإجمالي كالآتي:

total_income=bs+t+e

3- في البرنامج التالي توجد ثلاثة أخطاء استخرجها، مع تحديد سبب الخطأ، ثم أعد كتابة البرنامج بعد تصحيح الأخطاء:

```
void main()
{
int y=6
x=4;
y=y+x;
cout<<y;
}
```

بُنى التحكم Control Structures

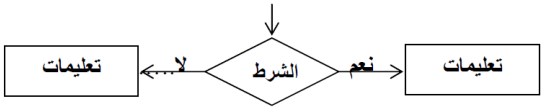
المقصود بذلك التركيبة التي تتحكم في مسار تنفيذ البرنامج، فإلى الآن يتم تنفيذ البرنامج بصورة تتابعية إلا إذا استخدمت هذه البنى ، وبنى التحكم تشمل الشرط والحلقات التكرارية.

بنية الشرط في C++:

الشرط هو ربط تنفيذ أمر معين "أو عدة أوامر" بتوفر شرط معين أو عدمه.

الصيغة العامة:

```
if(condition)
{
statements if true
……......
……
}
else
{
statements if false
……
……
}
```



نلاحظ من الصيغة العامة استخدام أداة الشرط if يعقبها الشرط ثم الأمر أو الأوامر المطلوب تنفيذها عند توفر هذا الشرط، لاحظ ان هذه الأوامر لن تنفذ إلا عند توفر الشرط وهذه الأجزاء إجبارية في بنية الشرط " أي يمكن أن نكتفي بالشرط والتعليمات التي تنفذ عند توفره دون طرح خيار آخر. ويمكن استخدام الأداة else لربط الأوامر التي يراد لها أن تنفذ عند انتقاء الشرط.

ملاحظات:

- ✓ عندما تكون الأوامر يجب استخدام الأقواس الحصرية {، } ويمكن الاستغناء عنها في حالة الأمر الواحد.
- ✓ تستخدم المعاملات العلائقية لصياغة الشرط وهي مبينة في جدول لاحق.
- ✓ عندما يكون الشرط مركباً تستخدم المعاملات المنطقية للربط بين الشروط.

المعاملات العلائقية Relational Operators:

وهي تحدد علاقة متغير بكمية ، أو متغير بمتغير آخر وهي ستة تشمل أكبر من ، أكبر من أو يساوي، أصغر من ، أصغر من أو يساوي، يساوي ، لا يساوي وهي موضحة في الجدول التالي:

الأداة	المعنى
>	أكبر من
>=	أكبرمن أو يساوي
<	أصغر من
<=	أصغر من أو يساوي
==	يساوي
!=	لا يساوي

كل هذه الأدوات تعمل في الاتجاه من اليسار إلى اليمين

مشكلة الخلط بين = و ==

الكثير من المبرمجين يخلطون بين معنيي هذين المعاملين، فمعامل الإسناد = يستخدم لنسبة كمية على يمينه إلى متغير على يساره، إذن فهذا المعامل لا يعني يساوي بل يعني أسند قيمة. أما == فهو معامل علائقي يعني يساوي ولا يستخدم إلا مع if .

مثال: عبر عن الشروط التالية بصيغة تناسب لغة C++.

1. إذا كانت x أكبر من y فإن d=5.
2. إذا كانت الدرجة d أكبر أو يساوي 60 اطبع ناجح "pass" وإلا اطبع راسب "fail".
3. إذا كانت قيمة الجذر التربيعي sr أقل من 0 اطبع عدد تخيلي "imaginary number".
4. إذا كانت a لا تساوي 5 فإن b=2a+3 ، و c=3a-4b، وإلا فإن b=2a-11 , c=3a+b.

الحل:

```
1. if(x>y)
   d=5;
2. if(d>=60)
       cout<<"pass";
   else
       cout<<"fail";
3. if(sr<0)
      cout<<" imaginary number";
4. if(a!=5)
   {
      b=2a+3;
      c=3a-4b;
   }
   else
  {
      b=2a-11;
      c=3a+b;
  }
```