Programming 1

C++ Numerical Data Input/Output

قواعد القسمة

C++ يتعامل مع الاعداد الصحيحة بشكل مختلف عن الاعداد العشرية

- 100 هو عدد من النوع int
- 100., 100.0000 , 100.0 هي اعداد من النوع double
- القواعد العامة لتقسيم int و double هي كالتالي:

– **double/double -> double (normal)**

– **double/int -> double (normal)**

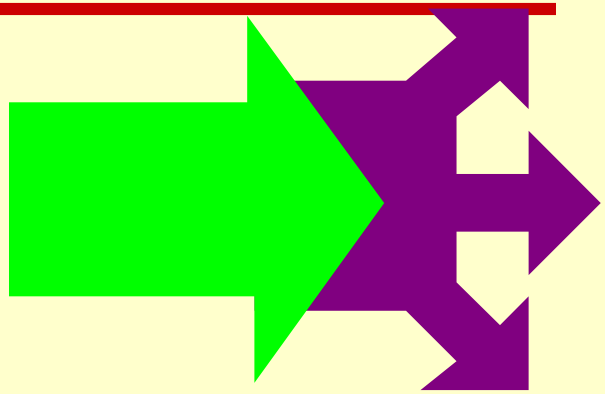
– **int/double -> double (normal)**

– **int/int -> int (ملاحظة: الجزء العشري سيتم تجاهله)**

- **220. / 100.0** `double/double -> double` **result is 2.2**
- **220. / 100** `double/int -> double` **result is 2.2**
- **220 / 100.0** `int/double -> double` **result is 2.2**
- **220 / 100** `int/int -> int` **result is 2**

Saul Greenberg

تحويل الاسناد (مثال)



```
Exmple1:
double yy = 2.7;
int i = 15;
int j = 10;
i = yy;                  // i is now 2
yy = j;                 // yy is now 10.0
Exmple2:
int m, n;
 double xx;
 m = 7;
 n = 2.5;
 xx = m / n;
 n = xx + m / 2;
                        // What is the value of n?
```

Saul Greenberg

نـوع المعطيات Enumerated Data Types

نوع من المعطيات يتم تعريفه من قبل المبرمج

يحتوى مجموعة من الأسماء الثابتة

الشكل العام

- enum name {val1, val2, ... valn};
- أمثلة:
- enum Fruit {apple, grape, orange};
- enum Days {Mon, Tue, Wed, Thur, Fri};
- لاستخدام هذه المعطيات يتم كالآتي:
- Days workDay, vacationDay;

Saul Greenberg

حل المثال

Example 2:

```
int m, n;
double xx;
m = 7;
n = 2.5;   // 2.5 converted to 2 and assigned to n
xx = m/n; // 7/2=3 converted to 3.0 and assigned to xx
n = xx+m/2;
  // m/2=3 : integer division
  // xx+m/2 : double addition because xx is double
  // convert result of m/2 to double (i.e. 3.0)
  // xx+m/2=6.0
  // convert result of xx+m/2 to int (i.e. 6)
  // because n is int
```

Saul Greenberg

إجبار تغيير نوع المتحول (Forcing a Type Change)

يمكن تغيير نوع متحول أو تعبير بواسطة الصيغة العامة التالية:

```
variable1 = type(variable2);
variable1 = type(expression);
```

Example:

```
int x=1, y=2;
double result1 = x/y;                    // result1 is 0.0
double result2 = double(x)/y;            // result2 is 0.5
double result3 = x/double(y);            // result3 is 0.5
double result4 = double(x)/double(y);    // result4 is 0.5
double result5 = double(x/y);            // result5 is 0.0
int cents = int(result4*100);            // cents is 50
```

Saul Greenberg

تنسيق صيغة الاخراج

```
#include <iomanip>
```

- **setw(int size)**

يحدد عدد الأحرف المراد استخدامها في عرض القيمة اللاحقة

Ex: cout << setw(5) << 12; //output 3 spaces and then 12

- **setprecision(int digit)**

يحدد عدد الأرقام المهمة لجميع المخرجات اللاحقة.

Ex: **Double pi=3.1415926**

cout << setprecision(4)<<pi; // 3.142

- **cout << fixed << setprecision(2)<<pi;** // 3.14

الدقة هي: رقمين بعد العلامة العشرية.

Saul Greenberg

بـعض أحرف الإخراج الخاصة:

- `\a` bell
- `\t` tab
- `\n` new line
- `\'` single quote
- `\"` double quote