Chapter 4

Mathematical Functions, Characters, and Strings

Section 4.1–4.11

More with char type:

- Escape sequence:

- يوجد بعض الأحرف المميزة الي بتأدي وظيفة محددة وتستخدم في جمل الطباعة ، مثل :

Escape sequence	description
\b	بتحذف الحرف الي قبله
\t	بتترك مسافة
\n	بتنزل سطر جديد
\r	بترجع لأول حرف
\\	بتطبع \
\"	بتطبع "
\'	بتطبع '

Examples:

```
cout << "backSpacee\b ";
```
Output: backSpace

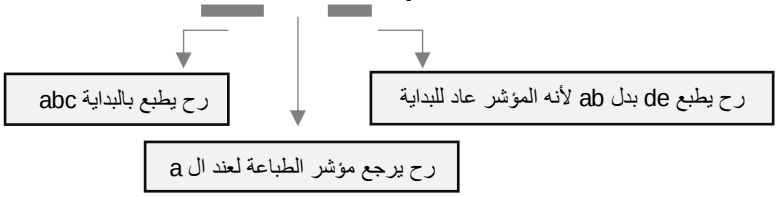
```
cout << "Tab\tTab ";
```
Output: Tab Tab

```
cout << "Quote: \'abc\' ";
```
Output:'abc'

```
cout << "He said \"Hi!\" ";
```
Output: "Hi!"

```
cout << "newLine \n newLine ";
```
Output : newLine
newLine

```
cout << "abc\rde";
```
Output: dec



- Character Functions :

الـ functions هي كودات جاهزة موجودة باللغة بتأدي وظيفة محددة ورح نتعرف عليهم لاحقا، لكن هناك بعض الـ function الي بتتعلق بال characters ، منها :

Function	Description: ch is a char variable
isdigit(ch)	برجع true اذا هو رقم
isalpha(ch)	برجع true اذا هو حرف
isalnum(ch)	برجع true اذا هو حرف او رقم
islower(ch)	برجع true اذا الحرف small letter
isupper(ch)	برجع true اذا الحرف capital letter
isspace(ch)	برجع true اذا الحرف space
tolower(ch)	بخلي الحرف small letter
toupper(ch)	بخلي الحرف capital letter

بنستخدم هاي ال functions مع شرط ال if لانها **بترّجع** true او false

بعض الأمثلة حتى نوضح الفكرة :

```
1) char x = 'a', y = 'B', z = '1';

isdigit(x);     //false          islower(x);     //true
isdigit(y);     //false          islower(y);     //false
isdigit(z);     //true           islower(z);     //false

isalpha(x);     //true           isupper(x);     //false
isalpha(y);     //true           isupper(y);     //true
isalpha(z);     //false          isupper(z);     //false

isalnum(x);     //true           isspace(x);     //false
isalnum(y);     //true           isspace(y);     //false
isalnum(z);     //true           isspace(z);     //false
```

كما نلاحظ انه قيمة الـ char تبقى كما هي هو فقط بعطيني true or false

```
if(isalpha(x))          if(true)          output:
  cout<<"Yes";    ⟶      cout<<"Yes";  ⟶   Yes
```

and x still = 'a'

- (tolower, toupper) هذول الـ funcitons بتحول حالة الحرف من **كبير لصغير** والعكس

```
char x = 'a';
x = toupper(x);
```

x = 'A'
خزنا القيمة على نفس الـ variable

```
char x = 'A';
char y = tolower(x);
```

x ='A', y = 'a'
خزنا القيمة ب y وقيمة x ما تغيرت

كما نلاحظ، انه الـ function ما بتعدل على قيمة الـ char نفسها كما في **الحالة الثانية**

الا اذا بدي اعدل على قيمة نفس الـ char بعمل زي **الحالة الأولى**

Example: write a program that promote a char from user and define if the char (upper, lower, digit):

```
#include <iostream>
using namespace std;
int main()
{

  char c;
  cout << "Enter a char: ";
  cin >> c;

  if (islower(c)) {
        cout << "The char " << c << " is lower \n"
        << "and the upper of " << c << " is " << (char)toupper(c);
  }
  else if (isupper(c)) {
        cout << "The char " << c << " is upper \n"
        << "and the lower of " << c << " is " << (char)tolower(c);
  }
  else if (isdigit(c))
        cout << "The char " << c << " is digit" << endl;
}
```

الـ toupper,tolower بعطوني رقم الحرف حسب الـ ascii table ، فعشان نطبع الحرف لازم نعمل casting

او بنخزنه ب char variable ولحاله بعمل implicit casting

Input:

Enter a char : a

Output:

The char a is lower and the upper of a is A

Input:

Enter a char : G

Output:

The char G is upper and the lower of G is g

- String type :

الـ string هي data type وبعبر عن **مجموعة** من الاحرف حتى تكون جملة او كلمة ولازم تكون بين " " ، وطريقة **انشاء** ال string كالتالي :

```
string s = "Hello c++!";
cout <<"The s = " << s << endl;
```

Output: The s = Hello c++!

بتتميز الـ string بإمكانية **الوصول** لكل حرف **والتعديل** عليه فهي تعتبر سلسلة من الاحرف المركبة على بعض :

```
string s = "Hello c++!";
```

index	0	1	2	3	4	5	6	7	8	9
value	H	e	l	l	o		c	+	+	!

s.length() is 10

s.at(0) or s[0]

s.at(9) or s[9]

- **index :** رقم بمثل الـ **location** الحرف ، وببدأ من (**0** الى **عدد الاحرف -1**)

ويوجد طريقتين للوصول الى ال index :

s[index]

```
cout << s[0];
```

s.at(index)

```
cout << s.at(0);
```

- دائما **index** اول حرف هو 0 و**index** اخر حرف هو s.size() -1

- لمعرفة **طول** او **عدد احرف** ال string بنستخدم ال function :

```
s.size()    or you can use    s.length()
```

- Reading Strings from user:

يوجد طريقتين لقراءة ال string من ال user :

1) Reading a word: **using** cin (كلمة واحدة فقط)

```
string s;
cout << "Enter a name:" << endl;
cin >> s;
cout << "Your name is: " << s;
```

```
Enter a name: Ayman Wail Qarout
Your name is: Ayman
```

بوقف قراءة عند اول space عشان هيك خزن اول كلمة

2) Reading a line : **using** getline(cin , var): (جملة كاملة)

```
string s;
cout << "Enter a full name:" << endl;
getline(cin, s);
cout << "Your name is: ";
```

```
Enter a name: Ayman Qarout
Your name is: Ayman Qarout
```

بوقف قراءة عند لما انزل سطر جديد عند الضغط على Enter

- Concatenating Strings:

تدعم الـ string عملية الـ +، وتؤدي الى دمج اكثر من **string** مع بعضهم البعض :

```
string s1="Hi", s2=" c++!";
string s3 = s1 + s2;
cout << s3;
```

Output: Hi c++!

or

```
string s = "Good";
s += " ";
s += "morning";
s += "!";
cout << s;
```

Output: Good morning!

- Formatting Console Output:

بالإمكان عمل تنسيق للـ output بحيث يكون اكثر **جمالية** ، عن طريق استخدام بعض الـ functions الموجودة في مكتبة **<iomanip>** :

Function	Description:	
setprecision(n)	بتتحكم بعدد الخانات للرقم المطبوع ، n : هي عدد الخانات ودائما الsetprecision موجودة وال default لل n هو 6	للأرقام
fixed	بتخلي تحكم ال setprecision بال floationg-point يعني n: هي عدد خانات ال floationg-point	
showpoint	اذا كان عدد الخانات المطبوع **اقل** من n فهي بتضيف 0 حتى تكمل الرقم	
setw(width)	بتعمل خانات للطباعة على حسب قيمة ال width By default start from right	للكلمات
left	بتيجي مع ال **setw** , بتخلي الطباعة تبدأ من **اليسار**	
right	بتيجي مع ال **setw** , بتخلي الطباعة تبدأ من **اليمين**	

Examples:

```
double num = 12.34567;
cout << setprecision(3) << num << endl
     << setprecision(4) << num << endl
     << setprecision(5) << num << endl
     << setprecision(6) << num << endl;
```

1)12.3 (3) 2)12.35 (4)
3)12.346 (5) 4)12.3457 (6)

لو نلاحظ اخر رقم بتم تقريبه على حسب الرقم الي قبله اذا بزيده 1 او لا

```
double num = 232123434.357;
cout << num << endl;
cout << fixed << num;
cout << fixed << setprecision(2) << num;
```

لما تكون الـ fixed لحالها بكون معها by default setprecision(6)

1) 2.32123e+08 → الصيغة العلمية للارقام الكبيرة

2) 232123434.357000 → طبعنا 6 ارقام بعد الفاصلة

3) 23212343.36 → طبعنا رقمين بعد الفاصلة

```
cout << setprecision(7);
cout << 1.23 << endl;
cout << showpoint << 1.23 << endl;
cout << showpoint << 123.0 << endl
```

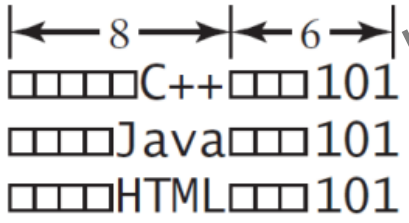
حددنا قيمة n بتساوي 7

لو ما كانت موجودة by default ال showpoint رح تضيفها والـ n بتساوي 6

1) 1.23 → showpoint طبع 3 ارقام لعدم وجود
2) 1.230000 → كمل الرقم ل 7 خانات
3) 123.0000 → كمل الرقم ل 7 خانات

```
cout << setw(8) << "C++" << setw(6) << 101 << endl;
cout << setw(8) << "Java" << setw(6) << 101 << endl;
cout << setw(8) << "HTML" << setw(6) << 101 << endl;
```

1) C++ 101
2) Java 101
3) html 101



ال setw عملت 8 خانات by default وبتبلش تعبي من **اليمين**

ال setw عملت 6 خانات by default وبتبلش تعبي من **اليمين**

بتخلي الـ setw تعبي من اليمين

```
cout << right;
cout << setw(8) << 1.23 << endl;
cout << setw(8) << 351.34 << endl;
```

□□□□1.23
□□351.34

دائمًا الـ setw بتتحدد حسب اخر right انكتبت. او left

by default بتكون محددة على الـ right

اذا ما كنا محددين أي وحدة منهم

بتخلي الـ setw تعبي من اليسار

```
cout << left;
cout << setw(8) << 1.23;
cout << setw(8) << 351.34 << endl;
```

□□□□1.23 □□351.34

- Mathematical Functions

يوجد في لغة ال c++ العديد من ال functions الرياضية الي رح تساعدنا في حل المعادلات الرياضية عن طريق استخدام مكتبة **<cmat>** :

- من هذه ال functions هي :

1) Trigonometric Functions :

Function	Description:
sin(radians)	بتحسب قيمة ال sin لل radians
cos(radians)	بتحسب قيمة ال cos لل radians
tan(radians)	بتحسب قيمة ال tan لل radians
asin(a)	بتحسب قيمة الزاوية الي sin تبعها بتساوي a
acos(a)	بتحسب قيمة الزاوية الي cos تبعها بتساوي a
atan(a)	بتحسب قيمة الزاوية الي tan تبعها بتساوي a

Examples:

```
cout << sin(0) << endl;
cout << sin(3.14 / 2) << endl;
cout << cos(0) << endl;
cout << atan(1) << endl;
```

```
0.0
1.0
1.0
0.7853
```

2) Exponent Functions :

Function	Description:
exp(x)	بتحسب قيمة e لقوة x ، e^x
log(x)	بتحسب قيمة ال log(x) للأساس e ، $ln(x)$
log10(x)	بتحسب قيمة log(x) للأساس 10 ، $log_{10}(x)$
pow(a,b)	بتحسب قيمة a أس b ، a^b
sqrt(x)	بتحسب قيمة الجذر التربيعي لل x ، $\sqrt{x}$

Examples:

```
cout << exp(1.0) << endl;
cout << log(E) << endl;
cout << log10(10) << endl;
cout << pow(2, 3) << endl;
cout << sqrt(4) << endl;
```

```
2.7
1.0
1.0
8
2
```

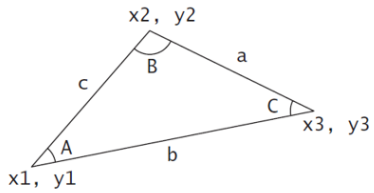
3) Service Functions :

Function	Description:
ceil(x)	بتعمل **تقريب** للرقم لأكبر عدد صحيح ، الجواب بكون double
floor(x)	بتعمل **تقريب** لأصغر عدد صحيح ، الجواب بكون double
min(x,y)	بترجع القيمة **الأصغر** بين x و y
max(x,y)	بترجع القيمة **الأكبر** بين x و y
abs(x)	بترجع القيمة **المطلقة** لل x

Examples:

```
ceil(2.1)  // 3.0     floor(2.1) // 2.0     min(2,3) // 2
ceil(-2.1) // -2.0    floor(-2.1) // -3.0    max(2.5,3.4) // 3.4
```

Examples: Computing Angles of a Triangle



$a = \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2}$	$A = cos^{-}((a^2 - b^2 - c^2) \div (-2 \times b \times c)$
$b = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2}$	$B = cos^{-}((b^2 - a^2 - c^2) \div (-2 \times a \times c)$
$c = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$	$C = cos^{-}((c^2 - b^2 - a^2) \div (-2 \times a \times b)$

A program that prompts the user to enter the x- and y-coordinates of the three corner points in a triangle and then displays the triangle's angles (A,B,C).

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
	// Prompt the user to enter three points
	cout << "Enter three points: ";
	double x1, y1, x2, y2, x3, y3;
	cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3;

	// Compute three sides
	double a = sqrt((x2 - x3) * (x2 - x3) + (y2 - y3) * (y2 - y3));
	double b = sqrt((x1 - x3) * (x1 - x3) + (y1 - y3) * (y1 - y3));
	double c = sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));

	// Obtain three angles in radians
	double A = acos((a * a - b * b - c * c) / (-2 * b * c));
	double B = acos((b * b - a * a - c * c) / (-2 * a * c));
	double C = acos((c * c - b * b - a * a) / (-2 * a * b));

	// Display the angles in degress
	const double PI = 3.14159;
	cout << "The three angles are " << A * 180 / PI << " "
		<< B * 180 / PI << " " << C * 180 / PI << endl;
}
```

دائما جواب العملية الحسابية بكون بال radian

Radian to degress -> angle $\times \frac{\pi}{180}$

- Simple file output & simple file input :

تعلمنا عن عمليات الإدخال والإخراج على **الشاشة** باستخدام مكتبة الـ <iostream>

ورح نتعلم كيف نعمل عمليات ادخال وإخراج على **file** باستخدام مكتبة الـ <fstream>

- عملية الكتابة او الطباعة على الملف باستخدام (**ofstream**) :

اسم الـ variable الي رح نطبع من خلاله على الملف

```
ofstream output("file_name.txt");

output << 123 << " Hi !" << endl;
```

هون بنحط اسم الملف ولازم نرفق معه .txt

نفس فكرة الـ cout لكن بنستخدم اسم الـ variable

- عملية الادخال او اخذ الـ data من الملف باستخدام (**ifstream**) :

- على سبيل المثال لو كان عنا بداخل الملف ما يلي : (123 1.2 Hi c++)

اسم الـ variable الي من خلاله رح ناخذ القيم من الملف

```
ifstream input("file_name.txt");

int x;  dobule y;  string s1, s2;

input >> x >> y >> s1 >> s2;
```

هون بنحط اسم الملف ولازم نرفق معه .txt

x = 123 y =1.2

s1 = "Hi" s2 = "c++"

نفس فكرة الـ cin لكن بنستخدم اسم الـ variable