if-else statement: if (...) {...}
else if (...) {...}
else {...}

```
if (num < 5)
   cout << "The number is less than 5." << endl;
else if (num > 10)
   cout << "The number is greater than 10." << endl;
else
   cout << "The number is in range [5, 10]." << endl;
```

?: operator

```
if(isPositive)
   factor = 1;
else
   factor = -1;
```

↓

```
factor = isPositive ? 1 : -1;
```

↓

```
factor = (isPositive) * 2 - 1;
```

A ? B : C (false / true)

在C/C++中, true=1, false=0, 故可用下面这种方法.

if后括号里的条件必须是可以转换为bool的. 其值可以为 bool, char, int, float.
(但一般只用 bool 和 int)

for loop: for (initialization; test-expression; update-expression) { body; }

C和C++不会报数组越界, 越界后, 内存有什么, 就取出什么.

C++11 标准提出:
foreach 迭代器

```
for (double x : prices)
   cout << x << std::endl;
```

→ x取出一份拷贝

```
int i = 0;
for (double &x : prices)
{
   x = x * 0.80; //20% off sale
   cout << x <<"  "<< prices[i++]<<endl;
}
```

→ 表示地址索引, 对其修改也会改变原数组

while loop:
```
init-expression;
while (test-expression)
{
   statements;
   update-expression;
}
```

do-while loop:
```
init-expression;
do
{
   statements;
   update-expression;
} while (test-expression)
```

```
int sum = 0;
for(int i = 0; i < 10; i++)
{
   sum += i;
   cout << "Line " << i << endl;
}
```

→

```
int sum = 0;
int i = 0;
while (i < 10)
{
   sum += i;
   cout << "Line " << i << endl;
   i++;
}
```

```
while(num > 0)
{
   cout << "num = " << num << endl;
   num--;
}
```

→

```
for(; num > 0; )
{
   cout << "num = " << num << endl;
   num--;
}
```

Endless loop

```
for(;;)
{
   // some statements
   cout << "endless loop!" << endl;
}
```

```
while(true)
{
   // some statements
   cout << "endless loop!" << endl;
}
```

goto (不推荐)

```
float mysquare(float value)
{
   float result = 0.0f;

   if(value >= 1.0f || value <= 0)
   {
      cerr << "The input is out of range." << endl;
      goto EXIT_ERROR;
   }
   result = value * value;
   return result;

EXIT_ERROR:
   //do sth such as closing files here
   return 0.0f;
}
```

- Execute one of several statements, depending on the value of a condition
- break prevents to execute some following statements. **Don't' forget break!**
- More similar with goto, not if-else if-else

switch.cpp

```
switch (input_char)
{
   case 'a':
   case 'A':
      cout << "Move left." << endl;
      break;
   case 'd':
   case 'D':
      cout << "Move right." << endl;
      break;
   default:
      cout << "Undefined key." << endl;
      break;
}
```

switch statement: switch(integer-expression)

```
{
    case label1 : statements;
    case label2 : statements;
    ...
    default     : statements;
}
```

一般在statements最后都要加break.

break：停止循环

continue：跳过当前循环，进行下次循环

若在多重循环中，break/continue只会对最内层循环作用，若想作用外层循环，只能在外层多加一个break/continue.

switch更类似于goto，而非if-else if-else.

Compare Operator：==、>、<、!=、>=、<=

数组字符串比较：strcmp(,)

数组字符串比较不能用==等，因为用比较符比较的是地址

在c++的string封装类中，可以用比较符比较，比较的是内容.(因为重载了运算符)

cctype库中有一些字符函数：isalnum()，isdigit()，isspace()

是否全是数字　是否数字　是否空格

Relational Expression

Operator name	Example
equal to	a == b
not equal to	a != b
less than	a < b
greater than	a > b
less than or equal to	a <= b
greater than or equal to	a >= b

当条件正确时返回1，否则返回0.

Logical Expressions：返回bool值

Operator name	Symbol-like operator	Keywork like operator	Example
negation	!	not	!a
AND	&&	and	a && b
Inclusive OR	||	or	a || b

优先级：! > && > ||

若操作数不是bool，会被隐式地转为bool.

- They will be converted to bool implicitly if it is feasible.

```
float count = 0.2f;
if (count) //not recommend to use a float-point number
    cout << "There are some." << endl;
```

- Pointers are also frequently used as conditions

```
int * p = new int[1024];
if (!p) // if(p == NULL)
    cout << "Memory allocation failed." << endl;
```