Advanced Programming in C++

فرشاد حکیم پور

Type Conversion (casting)

```
int i = 65;
double d = 5.0;
// Automatic (implicit) conversion
d = i;
// Explicit conversion to ‘A’
std::cout<<static_cast<char>(d)<<std::endl;
```

C++ Arithmetic Operators

- +
  `6 + 2     // 8`
- -
  `6 – 2     // 4`
- *
  `6 * 2     // 12`
- /
  `6 / 2     // 3`
- %
  `6 % 2     // 0`

C++ assignment

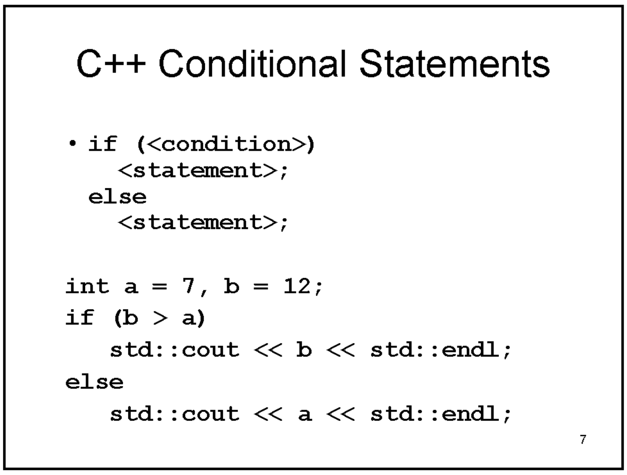
```
• a = b;
• a = b + c;
• a += b;       //a = a + b;
• a *= b + c;   //a = a * (b + c);
• a = b + (c = 5); //c = 5; a = b + c;
• i++;    // i = i + 1; or ++i;
• i--;    // i = i - 1; or --i;
• j = i++ // j = i; i = i + 1;
• j = ++i // i = i + 1; j = i;
```

C++ Comparison Operators

- <
  `6 < 2    // false`
- >
  `6 > 2    // true`
- <=
  `6 <= 2   // false`
- >=
  `6 >= 2   // true`
- ==
  `6 == 2   // false`
- !=
  `6 != 2   // true`

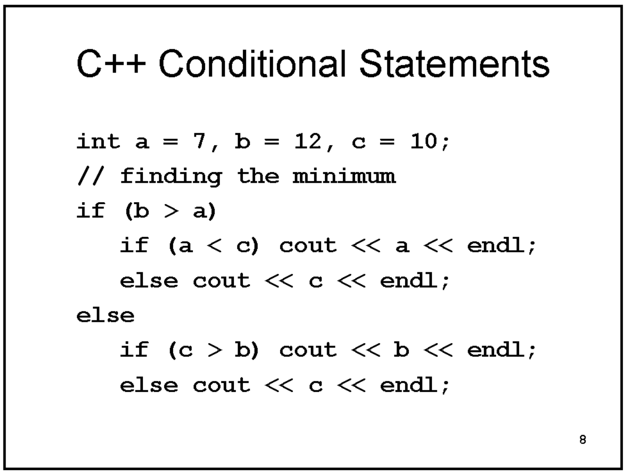
C++ Conditional Statements

```
• if (<condition>)
    <statement>;

int a = 7, b = 12;
int max = a;
if (b > max) max = b;
std::cout << max << std::endl;
```



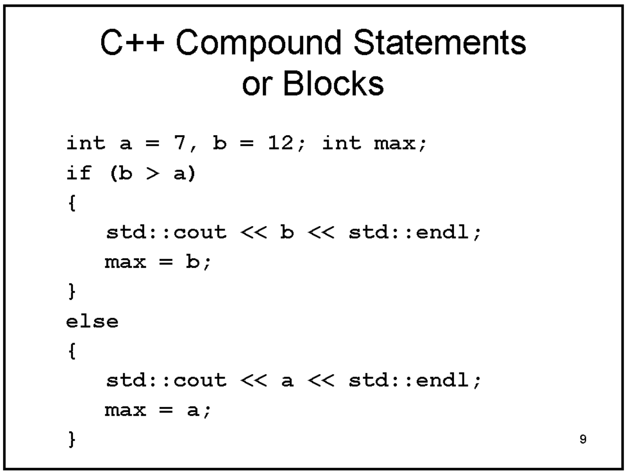
C++ Conditional Statements

```
• if (<condition>)
     <statement>;
  else
     <statement>;

int a = 7, b = 12;
if (b > a)
   std::cout << b << std::endl;
else
   std::cout << a << std::endl;
```



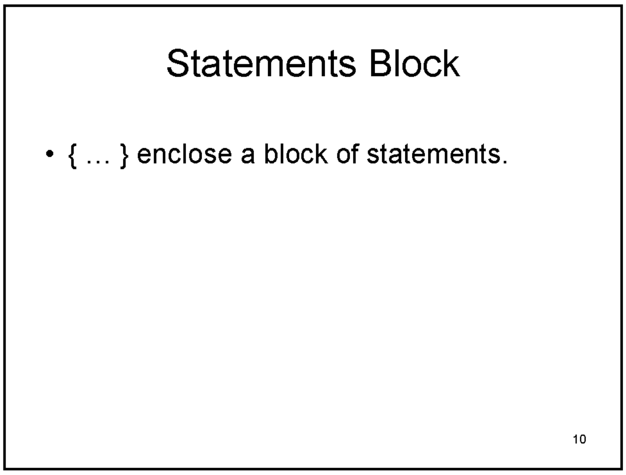
C++ Conditional Statements

```
int a = 7, b = 12, c = 10;
// finding the minimum
if (b > a)
   if (a < c) cout << a << endl;
   else cout << c << endl;
else
   if (c > b) cout << b << endl;
   else cout << c << endl;
```



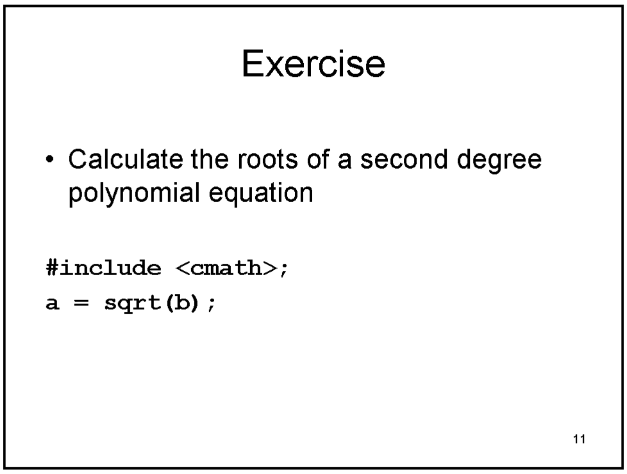
C++ Compound Statements or Blocks

```
int a = 7, b = 12; int max;
if (b > a)
{
   std::cout << b << std::endl;
   max = b;
}
else
{
   std::cout << a << std::endl;
   max = a;
}
```



Statements Block

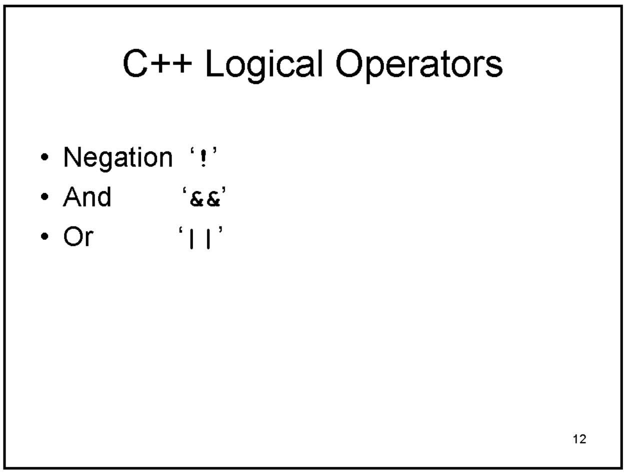
- { … } enclose a block of statements.



Exercise

- Calculate the roots of a second degree polynomial equation

```
#include <cmath>;
a = sqrt(b);
```



C++ Logical Operators

- Negation ‘!’
- And ‘&&’
- Or ‘||’

C++ Bitwise Operators

- And 	‘&’
- Or 	‘| ’
- Exclusive Or ‘^’
- Right Shift ‘>>’
- Left Shift ‘<<‘
- Negation ‘~’ (One’s Complement)

C++ Loops

- `while`
- `for`
- `do … while`

while

```
while (<condition>)
    <statement>;

int k = 5;
double factorial = 1;
while (k > 1) //factorial *= k--;
{
   factorial = factorial * k;
   k--;   // --k;
}
Std::cout << factorial << std::endl;
```

for

```
for (<init-statement>; <condition>; <expression>)
    <statement>;

int k = 5;
double factorial = 1;
for (int i = 2; i <= k; ++i)
   factorial *= i;
cout << factorial << endl;
```

do … while

```
do
 <statement>;
while (<condition>);

int k = 5;
double factorial = 1;
do
   factorial *= k;
   k--;
While (k > 1)
cout << factorial << endl;
```

“`break`” and “`continue`”

- “`break`” exits a loop
- “`continue`” jumps to the next iteration

Exercise

- Calculate greatest common divisor (gcd) of two numbers
- Calculate least common multiple (lcm) of two numbers