

Summary of basic C++-commands

K. Vollmayr-Lee, O. Ippisch

April 13, 2010

1 Compiling

To compile a C++-program, you can use either `g++` or `c++`.

```
g++ -o executable_filename.out sourcefilename.cc
c++ -o executable_filename.out sourcefilename.cc
```

Each command in C++ is followed by “;”. Carriage return has no meaning in C++. This means you can also distribute a command over several lines.

2 Comments

Essential for the writing of clear programs are comments, which are explanations for your program, and which are ignored by the compiler.

```
/* ... */ puts the text ... into comment (more than one line possible)
// ... puts text ... for rest of same line into comment
```

3 Data Types

| Data Type | Variable Declaration (Example) | Assignment (Example) | Range |
|----------------------|---|--------------------------------------|--|
| integer | short i1,i2; int i1,i2; long i1,i2; unsigned i1,i2; unsigned long i1,i2; | i1 = 3; | [−32768 : 32767] [−2.14 · 10 ⁹ : 2.14 · 10 ⁹] [−2.14 · 10 ⁹ : 2.14 · 10 ⁹] [0 : 4.29 · 10 ⁹] [0 : 4.29 · 10 ⁹] |
| real | float f1,f2; double f1,f2; long double f1,f2; | f1 = 3.2; f2 = 2.1E−3; | ±[10 ^{−38} : 10 ³⁸] 7 digits precision ±[10 ^{−308} : 10 ³⁰⁸] 16 digits precision ±[10 ^{−4932} : 10 ⁴⁹³²] 19 digits precision |
| single character | char c1,c2; | c1 = 'R' | |
| string of characters | string s1,s2; | s1 = "Farmer" | |
| logical | bool b1,b2; | b1 = true ; b2 = false | |

3.1 Constants

A constant value is defined with:

```
const type variableName = value;
```

The value of a constant can not be altered after the declaration.

Examples:

```
const int numPoints = 10;
const double epsilon = 1E-20;
```

4 Input And Output

4.1 Input/Output With Screen:

To be able to use the following commands you need to write

```
#include <iostream>
using namespace std;
```

at the beginning of your program. If you leave out the line **using namespace std**;
you have to use `std::cout`, `std::endl` and `std::cin` instead of `cout`, `endl` and `cin`.

4.1.1 output

```
cout << "string_of_characters"
cout << variable << endl;
```

4.1.2 input

```
cin >> variable;
```

4.2 Input/Output With Files:

To be able to use the following commands you need to write

```
#include <fstream>
using namespace std;
```

at the beginning of your program. If you leave out the line **using namespace std**;
you have to use `std::ofstream`, and `std::ifstream` instead of `ofstream` and `ifstream`.

4.2.1 output

```
ofstream outfilevariable ("outputfilename", ios::out);
outfilevariable << value;
```

will write the value into the file with name `outputfilename`.

4.2.2 input

```
ifstream infilevariable ("inputfilename", ios::in);
infilevariable >> a;
```

will read a value from the file with name `outputfilename` and store it in the variable `a`.

5 Arithmetic Calculations

5.1 Operations

+ - * /

5.2 Mathematical Functions

To be able to use all of the following functions you need to write at the beginning of your program:

```
#include <cmath>
```

| C++ name | function |
|-----------|---|
| pow(x,y) | x^y |
| sin(x) | |
| cos(x) | |
| tan(x) | |
| asin(x) | $\sin^{-1}(x)$ in range $[-\pi/2, \pi/2]$ |
| acos(x) | $\cos^{-1}(x)$ in range $[0, \pi]$ |
| atan(x) | $\tan^{-1}(x)$ in range $[-\pi/2, \pi/2]$ |
| sinh(x) | |
| cosh(x) | |
| tanh(x) | |
| exp(x) | e^x |
| log(x) | $\ln(x)$ |
| sqrt(x) | \sqrt{x} |
| fabs(x) | $ x $ |
| floor(x) | largest integer not greater than x; example: floor(5.768) = 5 |
| ceil(x) | smallest integer not less than x; example: ceil(5.768) = 6 |
| fmod(x,y) | floating-point remainder of x/y with the same sign as x |
| x \% y | remainder of x/y , both x and y integers |

6 Decision Statements

6.1 Comparison Operators

| C++ name | function | example |
|----------|----------|--------------------------|
| == | = | i1 == i2 |
| != | ≠ | i1 != i2 |
| > | > | i1 > i2 |
| < | < | i1 < i2 |
| >= | ≥ | i1 >= i2 |
| <= | ≤ | i1 <= i2 |
| && | and | (i1 != i2) && (i1 == i3) |
| | or | (i1 == i2) (i1 == i3) |

Be careful: using the assignment = instead of the comparison == is one of the very common errors in C++.

6.2 Statements

```
if( condition )  
{  
    statements  
}
```

```
if( condition )
```

```

{
    statements
}
else
{
    statements
}

if( condition )
{
    statements
}
else if
{
    statements
}
else
{
    statements
}

switch ( casevariable )
{
    case value1a:
    case value1b:
        {statements}
        break;
    case value2a:
    case value2b:
        {statements}
        break;
    default:
        {statements}
}

```

6.3 Repetitions

```

while ( conditions )
{
    statements
}

for ( init; conditions; update )
{
    statements
}

do
{
    statements // these statements are done
               // before the while statement is checked
} while ( condition );

```

7 Functions

A function is a set of commands. It is useful to write a user-defined function, i.e. your own function, whenever you need to do the same task many times in the program. All programs start execution at the function main. Three steps are important in the use of functions:

1. Before a function is used it has to be declared:

```
function_type function_name(types_of_parameter_list);
```

```
double feetinchtometer(int ,double );  
void metertofeetinch(double , int &, double &);
```

The function type declares which variable is passed back from the function (**void** means none). The variables *without* “&” are all input parameters, i.e. only passed to the function and are not changed within the function. The variables *with* “&” may be passed both to and from the function and may be changed in the function.

2. In the program you use the function with:

```
function_name(actual_parameter_list);
```

Examples:

```
feetinchtometer(5.0,3.2);  
metertofeetinch(1.3,feet,inch);
```

3. At some place the function has to be defined with:

```
function_type function_name(parameter_types_and_names)  
{declarations and statements }
```

If a function is defined before its first usage, the declaration (see above, number 1) is unnecessary. Examples:

```
double feetinchtometer(int feet , double inch)  
{...};  
  
void metertofeetinch (double m, int &feet , double &inch)  
{...};
```