

Multimedia Programming 2004

Erwin M. Bakker
Joachim Rijsdam

C++

- C designed and implemented by Dennis M. Ritchie (1969-1973 at Bell Labs)
- ANSI C standard definition (1983-1989)
- C++
 - Bjarne Stroustrup (1983-1985 at Bell Labs)
 - ANSI standard November 1997
- 'C with classes' suitable for object oriented programming

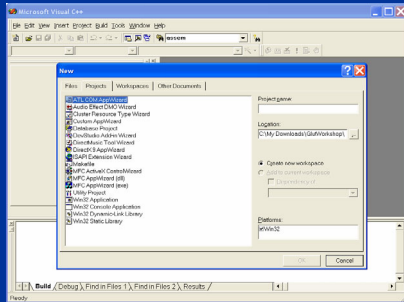
C++

- C is close to the machine => fast code
- C language constructs similar to that of other languages => Pascal, Visual Basic, C#, JAVA, Fortran, Ada almost for free
- Other programming and scripting languages like Perl, Python, LISP, CGI, PHP, will be easier to learn (Note: the other way around is not always true.)
- Lots of libraries available: you can program from scratch, but you can also stand on the shoulders of giant statues.....

C++ Skeleton Code

- the task that you have to program is probably, partly, already done before
- Find the right libraries, or code examples that you can adapt for your own task
- C++ Skeleton code
- C++ Libraries
- DirectX Skeleton code

MS Visual C++



Other C++ Environments

- Unix: gcc
- Borland C++ Builder
- Watcom C++
- Etc.
- Portability
 - ANSI standard
 - Proprietary Libraries
 - GUI resources
 - Make/build settings
 -

C++ Compiling

- 1.C++ code
- 2.Compilation
- 3.Object code
- 4.Linking
- 5.Executable

I/O Functions

- Formatted I/O <stdio.h>
 - printf("Text with format control %s",stringvar)
 - scanf("%d",integervar)
- Output streams <iostream.h>
#include <iostream.h>
...
int number1 = 0, number2 = 0;
cout << "Hello World\n";
cin >> number1 >> number2;
cout << "Product: " << number1

Expressions and Statements

Expressions

- `a+b` `1 x=p+d`
- `38/5` `39%4` (is equal to mod)
- `a = a + 1` `a += 1` `a *=3`
- `a++` `++a` `a--` `--a`

Statements

- `i=3; b = (i++); c = (++i);`
(=> `b = 3` and `c = 5`)

Types and Variables

■ <code>char</code>	(1 byte)
■ <code>short</code> (short int)	(2 byte)
■ <code>int</code>	(2 or 4 bytes)
■ <code>enum</code>	(2 or 4 bytes)
■ <code>long</code> (long int)	(4 bytes)
■ <code>float</code>	(4 bytes)
■ <code>double</code>	(8 bytes)
■ <code>long double</code>	(10 bytes)

Types and Variables

- `int n=100;` declaration and definition
- `extern int n;` declaration only
- `static int n=9;` `n` will not change anymore
- `float number;`
- `enum days { Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}`
`today;`
`today = Monday;`

Relational and Logic Operators

<code><</code>	less than	
<code>></code>	greater than	
<code><=</code>	<code>>=</code>	1 = true
<code>==</code>	equal to	0 = false
<code>!=</code>	unequal to	
<code>&&</code>	logical AND	
<code> </code>	logical OR	
<code>!</code>	logical NOT	

Conditional Statements

if (expression) statement else statement

Example:

```
if (x >= 0)
{
    printf("x is positive\n");
}
else
{
    printf("x is negative\n");
}
```

Conditional Statements

Clear:

```
if (x >= 0)
    printf("x is positive\n");
else
    printf("x is negative\n");
```

Not so clear:

```
if (true)
if (false)
    printf("one");
else
    printf("two");
```

Iterative Statements: while, do, for

- while (expression) do statement
- do statement while (expression);
- for (expr1; expr2; expr3) statement

```
int i = 0;
int sum = 0;
while (i < 10)
{
    sum += 1;
    i += 1;
}
```

Iterative Statements: while, do, for

- while (expression) do statement
- do statement while (expression);
- for (expr1; expr2; expr3) statement

```
int i = 0;
int sum = 0;
do
{
    sum += 1;
    i += 1;
} while (i < 10)
```

Iterative Statements: while, do, for

- while (expression) do statement
- do statement while (expression);
- for (expr1; expr2; expr3) statement

```
int i = 0;
int sum = 0;
for (i = 0; i < 10; i++)
{
    sum += 1;
}
```

Switch Statement

- switch (expression) statement

```
char character;
switch (character)
{
    case 'x':
        printf("Stop\n");
        break;
    case 'y': case 'Y':
        printf("Yes\n");
        break;
    default:
        printf("No\n");
        break;
}
```