

Programming in C++ - Primer

Lesson 2 - Dive In

Jakub 'Eremiell' Marek
<marekj14@fel.cvut.cz>

Silicon Hill C++ Academy

2013/10/28

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

1 Revision

2 Formatting

3 Headers

4 I/O Streams

5 Variables

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Revisiting basics

C++ Primer

Jakub Marek

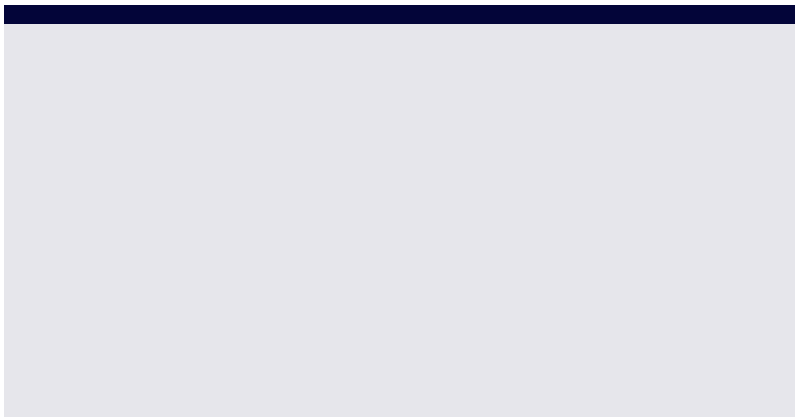
Revision

Formatting

Headers

I/O Streams

Variables



Revisiting basics

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Conditions

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Conditions

- if

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Conditions

- if
- switch/case

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Conditions

- if
- switch/case
- ternary

Revisiting basics

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Conditions

- if
- switch/case
- ternary

Loops

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Conditions

- if
- switch/case
- ternary

Loops

- while

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Conditions

- if
- switch/case
- ternary

Loops

- while
- do/while

Conditions

- if
- switch/case
- ternary

Loops

- while
- do/while
- for

Formatting

C++ Primer

Jakub Marek

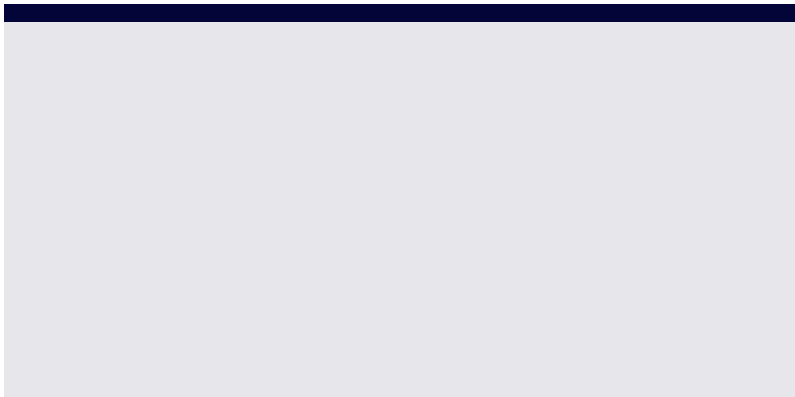
Revision

Formatting

Headers

I/O Streams

Variables



Formatting

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Compiler doesn't care about whitespace

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Compiler doesn't care about whitespace

- spaces

Compiler doesn't care about whitespace

- spaces
- tabulators

Compiler doesn't care about whitespace

- spaces
- tabulators
- newlines

Compiler doesn't care about whitespace

- spaces
- tabulators
- newlines

You can format your code in many ways

Compiler doesn't care about whitespace

- spaces
- tabulators
- newlines

You can format your code in many ways
Which can lead to unreadable code

Compiler doesn't care about whitespace

- spaces
- tabulators
- newlines

You can format your code in many ways
Which can lead to unreadable code

The International Obfuscated C Code Contest



<http://ioccc.org/>

Formatting styles

C++ Primer

Jakub Marek

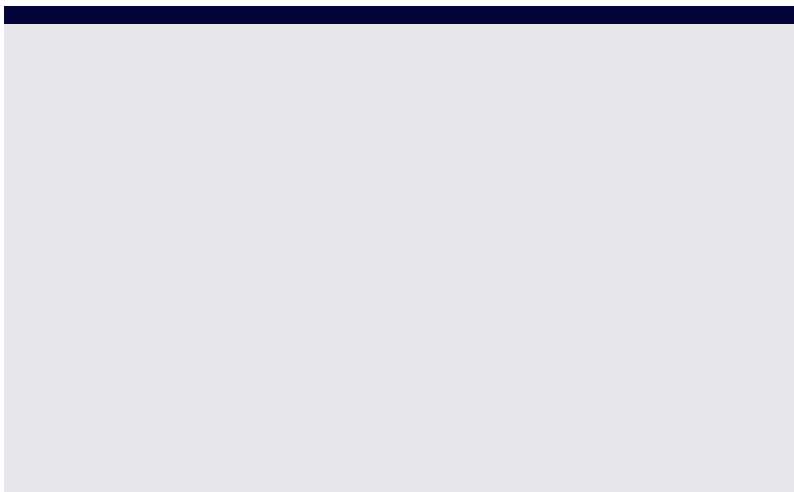
Revision

Formatting

Headers

I/O Streams

Variables



Formatting styles

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Allman (ANSI)

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Formatting styles

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Allman (ANSI)

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Java

```
int Foo(bool isBar) {
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Formatting styles

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Allman (ANSI)

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

K&R

```
int Foo(bool isBar)
{
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Java

```
int Foo(bool isBar) {
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```


Formatting styles

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Allman (ANSI)

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Java

```
int Foo(bool isBar) {
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

K&R

```
int Foo(bool isBar)
{
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Stroustrup

```
int Foo(bool isBar)
{
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Formatting styles

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Allman (ANSI)

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Java

```
int Foo(bool isBar) {
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

K&R

```
int Foo(bool isBar)
{
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Stroustrup

```
int Foo(bool isBar)
{
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Whitesmith

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Formatting styles

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Allman (ANSI)

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Java

```
int Foo(bool isBar) {
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

K&R

```
int Foo(bool isBar)
{
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Stroustrup

```
int Foo(bool isBar)
{
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Whitesmith

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Banner

```
int Foo(bool isBar) {
    if (isBar) {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Formatting styles

C++ Primer

Jakub Marek

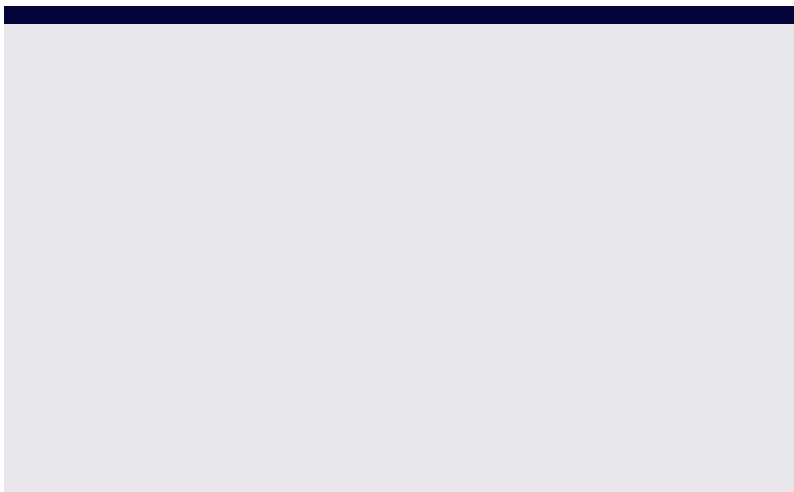
Revision

Formatting

Headers

I/O Streams

Variables



Formatting styles

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

GNU

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Formatting styles

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

GNU

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Linux

```
int Foo(bool isBar)
{
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Formatting styles

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

GNU

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Linux

```
int Foo(bool isBar)
{
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Horstmann

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Formatting styles

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

GNU

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Linux

```
int Foo(bool isBar)
{
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Horstmann

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

1TBS

```
int Foo(bool isBar)
{
    if (isFoo) {
        bar();
        return 1;
    } else {
        return 0;
    }
}
```


Formatting styles

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

GNU

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Linux

```
int Foo(bool isBar)
{
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Horstmann

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

1TBS

```
int Foo(bool isBar)
{
    if (isFoo) {
        bar();
        return 1;
    } else {
        return 0;
    }
}
```

Pico

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Formatting styles

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

GNU

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Linux

```
int Foo(bool isBar)
{
    if (isBar) {
        bar();
        return 1;
    } else
        return 0;
}
```

Horstmann

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

1TBS

```
int Foo(bool isBar)
{
    if (isFoo) {
        bar();
        return 1;
    } else {
        return 0;
    }
}
```

Pico

```
int Foo(bool isBar)
{
    if (isBar)
    {
        bar();
        return 1;
    }
    else
        return 0;
}
```

Lisp

```
int Foo(bool isBar) {
    if (isBar) {
        bar()
        return 1;
    }
    else
        return 0;
}
```

Compact formatting

C++ Primer

Jakub Marek

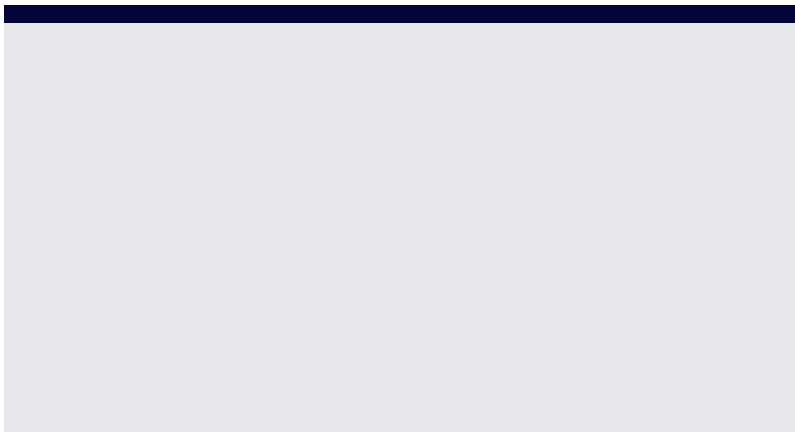
Revision

Formatting

Headers

I/O Streams

Variables



Compact formatting

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

```
int Foo(bool isBar) {  
    if (isBar) {  
        bar();  
        return 1;  
    }  
    else {  
        return 0;  
    }  
}
```

Compact formatting

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

```
int Foo(bool isBar) {  
    if (isBar) {  
        bar();  
        return 1;  
    }  
    else {  
        return 0;  
    }  
}
```

Tabs take less space and are harder to mishandle.

Questions?

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

C++ Primer

Jakub Marek

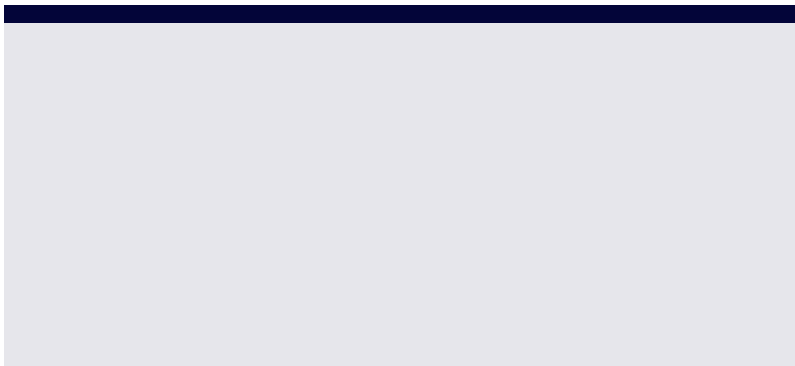
Revision

Formatting

Headers

I/O Streams

Variables



C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Two kinds of header files:

Two kinds of header files:

`<header>` System header files (libraries)

`"(path/)header.hpp"` Project header files

Two kinds of header files:

`<header>` System header files (libraries)

`"(path/)header.hpp"` Project header files

Dividing declarations and definition.

Two kinds of header files:

`<header>` System header files (libraries)

`"(path/)header.hpp"` Project header files

Dividing declarations and definition.

Cleaning up the code.

Two kinds of header files:

`<header>` System header files (libraries)

`"(path/)header.hpp"` Project header files

Dividing declarations and definition.

Cleaning up the code.

Reusability.

Two kinds of header files:

`<header>` System header files (libraries)

`"(path/)header.hpp"` Project header files

Dividing declarations and definition.

Cleaning up the code.

Reusability.

Every cpp got its own hpp.

Two kinds of header files:

`<header>` System header files (libraries)

`"(path/)header.hpp"` Project header files

Dividing declarations and definition.

Cleaning up the code.

Reusability.

Every cpp got its own.hpp.

Preprocessor variables come in all uppercase.

Preprocessor directives

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Preprocessor directives

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Executed before compilation

Preprocessor directives

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Executed before compilation

`#define` Defines a preprocessor variable, induces replaces.

`#undef` Undefines a preprocessor variable.

Preprocessor directives

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Executed before compilation

`#define` Defines a preprocessor variable, induces replaces.

`#undef` Undefines a preprocessor variable.

`#if` Checks condition (known to preprocessor).

`#elif` Checks another condition.

`#else` Other cases.

`#endif` Closes if block.

Preprocessor directives

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Executed before compilation

`#define` Defines a preprocessor variable, induces replaces.

`#undef` Undefines a preprocessor variable.

`#if` Checks condition (known to preprocessor).

`#elif` Checks another condition.

`#else` Other cases.

`#endif` Closes if block.

`#ifdef` Checks existence of definition.

`#ifndef` Checks inexistence of definition.

Preprocessor directives

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Executed before compilation

#define Defines a preprocessor variable, induces replaces.

#undef Undefines a preprocessor variable.

#if Checks condition (known to preprocessor).

#elif Checks another condition.

#else Other cases.

#endif Closes if block.

#ifdef Checks existence of definition.

#ifndef Checks inexistence of definition.

#include Includes another file (usually a header).

Preprocessor directives

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Executed before compilation

#define Defines a preprocessor variable, induces replaces.

#undef Undefines a preprocessor variable.

#if Checks condition (known to preprocessor).

#elif Checks another condition.

#else Other cases.

#endif Closes if block.

#ifdef Checks existence of definition.

#ifndef Checks inexistence of definition.

#include Includes another file (usually a header).

#line Changes number of line and name of file (for compiler only).

#error Prints out error and ends compilation.

#pragma Sets various compiler options.

Header shield

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Header shield

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

```
#ifndef HEADER_HPP
#define HEADER_HPP

#include <header1>
#include <header2>
#include "header3.hpp"

class c {
    ...
};

int func1(int);
double func2(char);

#endif
```

Questions?

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

I/O streams

C++ Primer

Jakub Marek

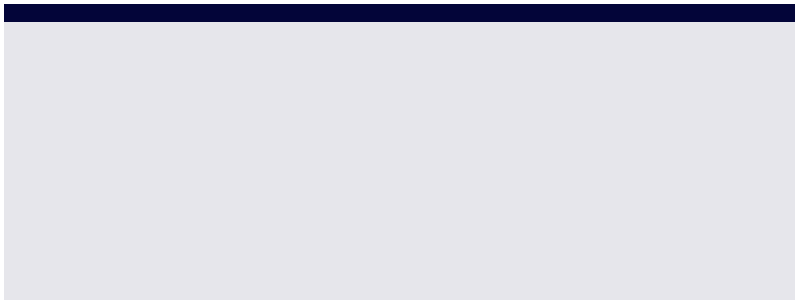
Revision

Formatting

Headers

I/O Streams

Variables



I/O streams

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

In UNIX, everything is a file.

I/O streams

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

In UNIX, everything is a file.
Devices are files too!

I/O streams

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

In UNIX, everything is a file.
Devices are files too!
C++ communicates using streams.

I/O streams

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

In UNIX, everything is a file.

Devices are files too!

C++ communicates using streams.

- Keyboard is standard input stream, `cin` & 0

I/O streams

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

In UNIX, everything is a file.

Devices are files too!

C++ communicates using streams.

- Keyboard is standard input stream, `cin` &0
- Terminal is standard output stream, `cout` &1

I/O streams

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

In UNIX, everything is a file.

Devices are files too!

C++ communicates using streams.

- Keyboard is standard input stream, `cin` &0
- Terminal is standard output stream, `cout` &1
- Terminal is also standard error stream, `cerr` &2

Standard input stream

C++ Primer

Jakub Marek

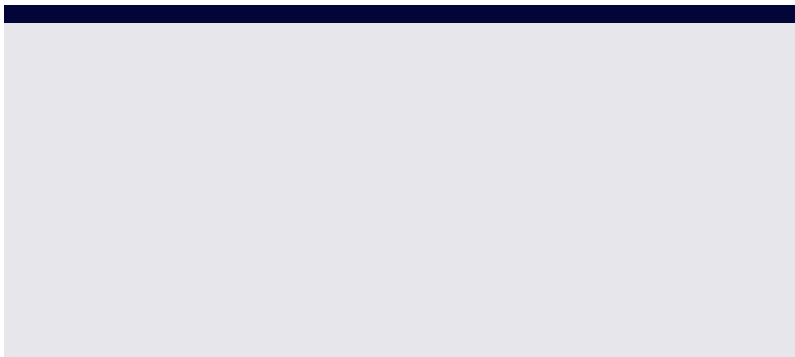
Revision

Formatting

Headers

I/O Streams

Variables



Standard input stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cin` is an instance of `istream` from `iostream` library.

Standard input stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cin` is an instance of `istream` from `iostream` library.

`operator»` Pushes data from stream to target variable.

Standard input stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cin` is an instance of `istream` from `iostream` library.

`operator»` Pushes data from stream to target variable.

`fail()` Checks for error in stream.

Standard input stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cin` is an instance of `istream` from `iostream` library.

`operator»` Pushes data from stream to target variable.

`fail()` Checks for error in stream.

`get()` Gets a character (or more) from stream.

Standard input stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cin` is an instance of `istream` from `iostream` library.

`operator»` Pushes data from stream to target variable.

`fail()` Checks for error in stream.

`get()` Gets a character (or more) from stream.

`getline()` Gets a line from stream.

Standard input stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cin` is an instance of `istream` from `iostream` library.

`operator»` Pushes data from stream to target variable.

`fail()` Checks for error in stream.

`get()` Gets a character (or more) from stream.

`getline()` Gets a line from stream.

`eof()` Checks if the stream is open.

Standard input stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cin` is an instance of `istream` from `iostream` library.

`operator»` Pushes data from stream to target variable.

`fail()` Checks for error in stream.

`get()` Gets a character (or more) from stream.

`getline()` Gets a line from stream.

`eof()` Checks if the stream is open.

`wcin` Wide stream for Unicode.

Standard output stream

C++ Primer

Jakub Marek

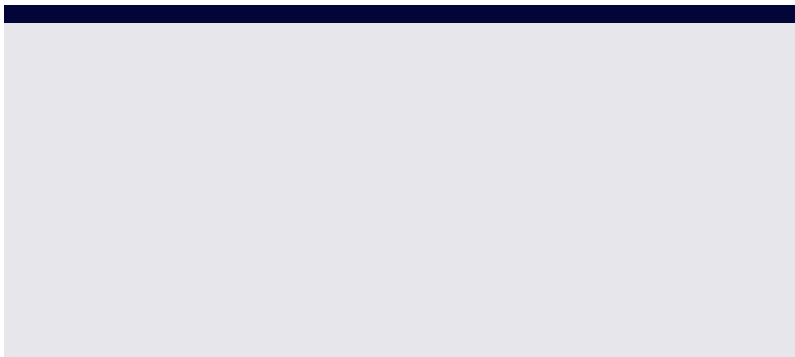
Revision

Formatting

Headers

I/O Streams

Variables



Standard output stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cout` is instance of `ostream` from `iostream` library.

Standard output stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cout` is instance of `ostream` from `iostream` library.
`operator«` Pushes data to stream.

Standard output stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cout` is instance of `ostream` from `iostream` library.

`operator«` Pushes data to stream.

`fail()` Checks for error in stream.

Standard output stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cout` is instance of `ostream` from `iostream` library.

`operator«` Pushes data to stream.

`fail()` Checks for error in stream.

`put()` Puts a character (or more) to stream.

Standard output stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cout` is instance of `ostream` from `iostream` library.

`operator«` Pushes data to stream.

`fail()` Checks for error in stream.

`put()` Puts a character (or more) to stream.

`write()` Puts a line to stream.

Standard output stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cout` is instance of `ostream` from `iostream` library.

`operator«` Pushes data to stream.

`fail()` Checks for error in stream.

`put()` Puts a character (or more) to stream.

`write()` Puts a line to stream.

`eof()` Checks if the stream is open.

Standard output stream

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

`cout` is instance of `ostream` from `iostream` library.

`operator«` Pushes data to stream.

`fail()` Checks for error in stream.

`put()` Puts a character (or more) to stream.

`write()` Puts a line to stream.

`eof()` Checks if the stream is open.

`wcout` Wide stream for Unicode.

Manipulators

C++ Primer

Jakub Marek

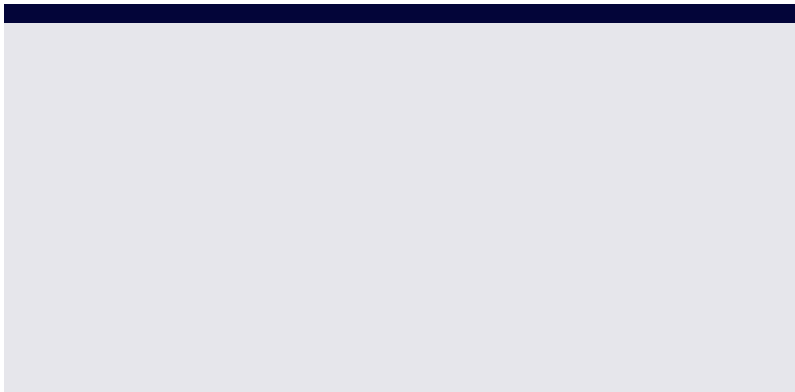
Revision

Formatting

Headers

I/O Streams

Variables



Manipulators

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Most in iomanip library

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Most in iomanip library

- endl

Manipulators

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Most in iomanip library

- endl
- dec, hex, oct

Manipulators

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Most in iomanip library

- endl
- dec, hex, oct
- fixed

Most in iomanip library

- endl
- dec, hex, oct
- fixed
- setw(<int>)
- setfill(<char>)

Most in iomanip library

- endl
- dec, hex, oct
- fixed
- setw(<int>)
- setfill(<char>)

```
std::cout « std::setiosflags(std::ios::uppercase);  
std::cout.flags(std::ios::uppercase);
```

Questions?

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Arrays

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Arrays

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Arrays are data structures, allowing several instances of one type under one identifier

Arrays

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Arrays are data structures, allowing several instances of one type under one identifier

Any dimensionality, just like tensors

<http://en.wikipedia.org/wiki/Tensor>

Arrays

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Arrays are data structures, allowing several instances of one type under one identifier

Any dimensionality, just like tensors

<http://en.wikipedia.org/wiki/Tensor>

Usually represent some kind of linked data

Arrays

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Arrays are data structures, allowing several instances of one type under one identifier

Any dimensionality, just like tensors

<http://en.wikipedia.org/wiki/Tensor>

Usually represent some kind of linked data

`<type><identifier>[<size>]...`

Arrays

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Arrays are data structures, allowing several instances of one type under one identifier

Any dimensionality, just like tensors

<http://en.wikipedia.org/wiki/Tensor>

Usually represent some kind of linked data

`<type><identifier>[<size>]...`

Size is fixed for the whole life of the array

Arrays

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Arrays are data structures, allowing several instances of one type under one identifier

Any dimensionality, just like tensors

<http://en.wikipedia.org/wiki/Tensor>

Usually represent some kind of linked data

`<type><identifier>[<size>]...`

Size is fixed for the whole life of the array

Arrays are indexed from 0, ending in size - 1

Arrays

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Arrays are data structures, allowing several instances of one type under one identifier

Any dimensionality, just like tensors

<http://en.wikipedia.org/wiki/Tensor>

Usually represent some kind of linked data

`<type><identifier>[<size>]...`

Size is fixed for the whole life of the array

Arrays are indexed from 0, ending in size - 1

Multidimensional arrays can be accessed both by several indices and by just one index

Arrays

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Arrays are data structures, allowing several instances of one type under one identifier

Any dimensionality, just like tensors

<http://en.wikipedia.org/wiki/Tensor>

Usually represent some kind of linked data

`<type><identifier>[<size>]. . .`

Size is fixed for the whole life of the array

Arrays are indexed from 0, ending in size - 1

Multidimensional arrays can be accessed both by several indices and by just one index

More on that later with pointers

Arrays

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Arrays are data structures, allowing several instances of one type under one identifier

Any dimensionality, just like tensors

<http://en.wikipedia.org/wiki/Tensor>

Usually represent some kind of linked data

`<type><identifier>[<size>]...`

Size is fixed for the whole life of the array

Arrays are indexed from 0, ending in size - 1

Multidimensional arrays can be accessed both by several indices and by just one index

More on that later with pointers

```
#include <iostream>
```

```
int main() {  
    int size = 5;  
    int a[size];  
    for (int i = 0; i < size; i++) {  
        a[i] = i;  
        std::cout << a[i] << std::endl;  
    }  
    return 0;  
}
```

Visibility of variables

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Visibility of variables

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Variable is always visible (and exists) only in the block, in which it was created (and all subblocks).

Visibility of variables

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Variable is always visible (and exists) only in the block, in which it was created (and all subblocks).

Variables created outside any block are called global, all the other are called local (to given block).

Visibility of variables

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Variable is always visible (and exists) only in the block, in which it was created (and all subblocks).

Variables created outside any block are called global, all the other are called local (to given block).

Global variables are considered bad design, especially in OOP (C++) and you shouldn't use them.

Visibility of variables

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Variable is always visible (and exists) only in the block, in which it was created (and all subblocks).

Variables created outside any block are called global, all the other are called local (to given block).

Global variables are considered bad design, especially in OOP (C++) and you shouldn't use them.

Variables can overshadow other variables with the same name.

Visibility of variables

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Variable is always visible (and exists) only in the block, in which it was created (and all subblocks).

Variables created outside any block are called global, all the other are called local (to given block).

Global variables are considered bad design, especially in OOP (C++) and you shouldn't use them.

Variables can overshadow other variables with the same name.

More local one always wins.

Visibility of variables

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Variable is always visible (and exists) only in the block, in which it was created (and all subblocks).

Variables created outside any block are called global, all the other are called local (to given block).

Global variables are considered bad design, especially in OOP (C++) and you shouldn't use them.

Variables can overshadow other variables with the same name.

More local one always wins.

```
#include <iostream>
```

```
int main() {  
    int a = 2;  
    std::cout << a << std::endl;  
    if (1) {  
        int a = 3;  
        std::cout << a << std::endl;  
    }  
    std::cout << a << std::endl;  
    return 0;  
}
```


Function overloading

C++ Primer

Jakub Marek

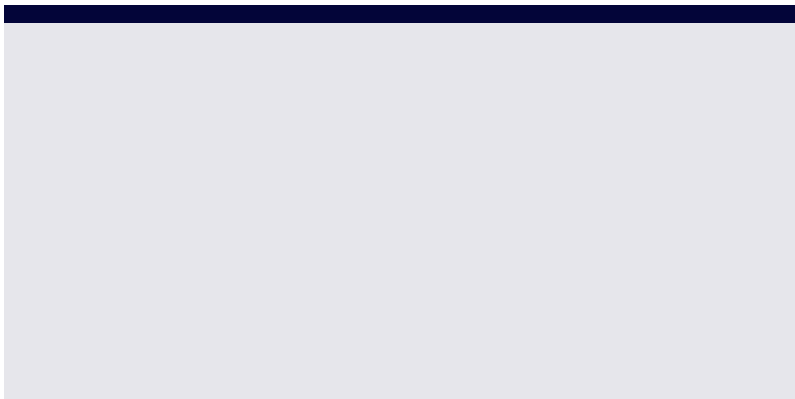
Revision

Formatting

Headers

I/O Streams

Variables



Function overloading

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Just as variables can be overshadowed, functions can be overloaded.

Function overloading

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Just as variables can be overshadowed, functions can be overloaded.
(Don't mix it up!)

Function overloading

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Just as variables can be overshadowed, functions can be overloaded.

(Don't mix it up!)

Overloading doesn't work on base of locality, but rather on difference of input types.

Function overloading

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Just as variables can be overshadowed, functions can be overloaded.

(Don't mix it up!)

Overloading doesn't work on base of locality, but rather on difference of input types.

```
int add(int , int );  
double add(int , double );  
double add(double , int );  
double add(double , double );
```

Questions?

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables

Break!

C++ Primer

Jakub Marek

Revision

Formatting

Headers

I/O Streams

Variables