

CS11600: Introduction to Computer Programming (C++)

Lecture 4

Svetlozar Nestorov
University of Chicago

Outline

- Introduction to C and C++
- "Hello World!" program
- Basic types
- Variables, constants and assignments
- Expressions and operators
- Control flow
- Input/Output (I/O)

1/12/2009

Svetlozar Nestorov, CS 116: Intro to Programming II

2

Introduction to C and C++

- The history of C and C++
 - <http://cm.bell-labs.com/cm/cs/who/dmr/chist.html>
 - C was devised in 70's at Bell Labs
 - Predecessors of C: BCPL, B.
 - Standardization: **ANSI C** in 1989.
 - C++ is the most widely used successor of C.
- Differences between C and C++

1/12/2009

Svetlozar Nestorov, CS 116: Intro to Programming II

3

"Hello World" Programs

- C program

```
#include <stdio.h>
int main() {
    int year = 2003;
    printf("Hello %d World!\n", year);
}
```
- C++ program

```
#include <iostream.h>
int main() {
    int year = 2003;
    cout << "Hello " << year << " World!" << endl;
}
```

1/12/2009

Svetlozar Nestorov, CS 116: Intro to Programming II

4

Built-in Types

- Integers (signed or unsigned):
`int`, `short`, `long`
- Real numbers (always signed):
`float`, `double`, `long double`
- Characters (signed or unsigned):
`char`
- Others:
`wchar_t`, `bool`

1/12/2009

Svetlozar Nestorov, CS 116: Intro to Programming II

5

Variables

- Variable definition:
`Type name [- value];`
- Examples:

```
int year = 2003;
int year;
char c = 'c';
double pi = 3.14;
```
- Note: variable name must start with a letter or `_` and cannot be a C++ *keyword*.

1/12/2009

Svetlozar Nestorov, CS 116: Intro to Programming II

6

More Variable Definitions

- Alternative initialization syntax:
`Type name(value);`
- Multiple variable definitions:
`Type name1(val1), name2(val2)...`
- Examples:
`int year = 2003, nextYear;`
`char c('c'), d='d', e;`

1/12/2009

Evgeniy Nazarov, CS 116: Intro to Programming II

7

Constants

- Constants (aka constant variables):
`const Type name = value;`
- Examples:
`const int year = 2003;`
`const char c = 'c';`
`const char c2 = c;`
- Note: constant must be initialized and this initial value cannot be changed.

1/12/2009

Evgeniy Nazarov, CS 116: Intro to Programming II

8

Assignments

- Basic form:
`name = expr;`
- `expr` can be a value (literal constant), constant, variable, or an expression.
- Another form:
`name1 = name2 = name3 = expr;`
- Assigns `expr` to all three variables.

1/12/2009

Evgeniy Nazarov, CS 116: Intro to Programming II

9

Basic Math Operators

- Arithmetic
`-, +, -, *, /, %`
- Logical
`! (not), && (and), || (or)`
- Bitwise
`~ (compl), & (bitand), | (bitor), ^ (xor)`
`<<, >>`
- Relational (comparisons)
`<, >, <=, >=, ==, != (not_eq)`

1/12/2009

Evgeniy Nazarov, CS 116: Intro to Programming II

10

Update Operators

- Update operators:
`+=, -=, *=, /=, %=, <<=, >>=, &=, |=, ^=`
- General form:
`name update_op expr;`
- Equivalent to:
`name = name op expr;`
- Examples:
`i += 5; (same as i = i + 5;)`
`c *= 10; (same as c = c * 10;)`

1/12/2009

Evgeniy Nazarov, CS 116: Intro to Programming II

11

Increment and Decrement

- There are 4 variations:
 - Prefix increment: `++name`
 - Prefix decrement: `--name`
 - Postfix increment: `name++`
 - Postfix decrement: `name--`
- Prefix updates the value *before* it is used in an expression while postfix updates it *after* it is used.
- Work for numbers and characters.

1/12/2009

Evgeniy Nazarov, CS 116: Intro to Programming II

12

More Operators

- Conditional operator
`cond ? expr1 : expr2`
- Evaluates to `expr1` if `cond` is true, otherwise evaluates to `expr2`
- Comma operator
`expr1, expr2, ...`
- Evaluates expressions left to right; value of operator is the last expression.

1/12/2022

Svetlana Nesterov, CS 116: Intro to Programming II

13

Control Flow

- Three kinds of control flow constructs:
 - Conditional: `if`, `if-else`, `switch`
 - Loop: `for`, `while`, `while-do`
 - Jump: `break`, `continue`, `goto`

1/12/2022

Svetlana Nesterov, CS 116: Intro to Programming II

14

If

- If has three forms:

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

```
if (condition) {  
    statements1;  
}  
else {  
    statements2;  
}
```

```
if (cond1) {  
    statements1;  
}  
else if (cond2) {  
    statements2;  
}  
-  
else {  
    statements3;  
}
```

1/12/2022

Svetlana Nesterov, CS 116: Intro to Programming II

15

Switch

- General form:

```
switch (expr) {  
    case constant1:  
        statements1;  
        break;  
    case constant2:  
        statements2;  
        break;  
    -  
    default:  
        statements1;  
        break;  
}
```
- Default is optional; order of cases is arbitrary, several cases may share statements.

1/12/2022

Svetlana Nesterov, CS 116: Intro to Programming II

16

While

- Two forms:

```
while (condition) {  
    statements;  
}
```

```
do {  
    statements;  
} while (condition)
```

- Evaluate condition, if true execute statements and repeat.
- Execute statements, evaluate condition, if true repeat.

1/12/2022

Svetlana Nesterov, CS 116: Intro to Programming II

17

For

- General form:

```
for (expr1; cond; expr3) {  
    statements;  
}
```
- Equivalent to:

```
expr1;  
while (cond) {  
    statements;  
    expr3;  
}
```

1/12/2022

Svetlana Nesterov, CS 116: Intro to Programming II

18