

Review: Exam 1

9/20/06

CS150 Introduction to Computer Science 1

1

Your First C++ Program

```
1 //*****
2 // File name: hello.cpp
3 // Author: Sheeren Khoja
4 // Date: 08/23/2006
5 // Purpose: This program displays a welcome message to
6 // the user
7 //*****
8 #include <iostream>
9 #include "stdafx.h"
10
11 using namespace std;
12
13 int main()
14 {
15     string name;
16
17     cout << "Type your name, then press enter" << endl;
18     cin >> name; /* read in the user's name */
19
20     cout << "Hello " << name << "!" << endl;
21
22     return 0;
23 }
24
```

What are the important parts of this program?
Keywords, variables, libraries, main function, input, output, comments.

9/20/06

CS150 Introduction to Computer Science 1

2

Declaration Statements

- Tells the compiler
 - The variable's type
 - The variable's name
- Examples of declaration statements

```
const double PI = 3.14;
double area;
char character = 'a';
int count;
bool bIsItASquare;
unsigned int bigNumber = 4000000000;
```

9/20/06

CS150 Introduction to Computer Science 1

3

Identifiers

- C++ does place limits on what names you can call your variables
- Rules
 1. Identifiers must begin with a letter or an underscore
 2. Identifiers must consist of letters, numbers and underscore, nothing else
 3. Identifiers cannot be a *reserved keyword*

9/20/06

CS150 Introduction to Computer Science 1

4

How to Choose a Data Type

- Ask yourself the following questions
 - Is the data a number or character?
 - Is the data a list of characters?
 - What are the largest and smallest numbers that may be stored?
 - How much memory does the variable use?
 - Is the variable signed (positive and negative)?
 - How many decimal places of precision does the variable need?

9/20/06

CS150 Introduction to Computer Science 1

5

Variable Ranges

Type	Size	Values
int	4 bytes	-2,147,483,648 to 2,147,483,647
short int	2 bytes	-32,768 to 32,767
long int	4 bytes	-2,147,483,648 to 2,147,483,647
unsigned int	4 bytes	0 to 4,294,967,295
char	1 byte	256 character values
float	4 bytes	1.2e-38 to 3.4e38
double	8 bytes	2.2e-308 to 1.8e308

9/20/06

CS150 Introduction to Computer Science 1

6

Arithmetic Operators

- Operators allow us to manipulate data

- Unary: **operator operand**
- Binary: **operand operator operand**
(left hand side) (right hand side)

Operator	Meaning	Type	Example
-	Negation	Unary	- 5
=	Assignment	Binary	rate = 0.05
*	Multiplication	Binary	cost * rate
/	Division	Binary	cost / 2
%	Modulus	Binary	cost % 2
+	Addition	Binary	cost + tax
-	Subtraction	Binary	total - tax

9/20/06

CS150 Introduction to Computer Science 1

7

Division

- grade = 100 / 40; grade is 2**
 - If both **operands** of the division **operator** are integers, then integer division is performed.
 - We say the integer is *truncated*. Everything after the decimal point is dropped. No rounding.
- grade = 100.0 / 40;**
 - grade** is 2.5
 - What data type should grade be declared as?

9/20/06

CS150 Introduction to Computer Science 1

8

Modulus

- Modulus is the remainder after integer division
- grade = 100 % 20;**
 - grade = ?**
- grade = 100 % 30;**
 - grade = ?**
- rem = x % n;**
 - What are the possible values for **rem**?

9/20/06

CS150 Introduction to Computer Science 1

9

Exponents

- The exponent operator was missing from the list! x^2 y^n
- C++ does not provide an exponent operator as part of the language
- Use `pow()` in the `cmath` library

```
#include <cmath>
double area;
area = pow(4, 2); // area = 42
```

9/20/06

CS150 Introduction to Computer Science 1

10

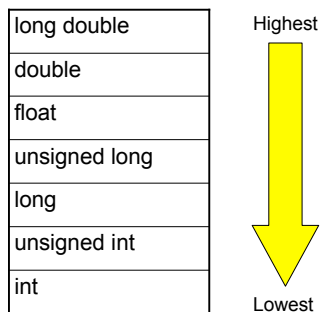
Implicit Type Conversion (3.3)

- What happens when we mix the data types of operands during mathematical operations
 - What happens when we save a double as an int?
 - What happens when an int is multiplied by a float?
- Data types are ranked
- A data type outranks another if it can hold a larger number

9/20/06

CS150 Introduction to Computer Science 1

11



9/20/06

CS150 Introduction to Computer Science 1

12

Rules for Type Conversion

- Rule 1: char, short, and unsigned short are automatically promoted to int
- Rule 2: When an operator works with values of different types, the lower ranking value is promoted to the higher ranking
- Rule 3: When the value of an expression is assigned to a variable, it is converted to the data type of that variable

9/20/06

CS150 Introduction to Computer Science 1

13

Uses of Type Casting

- Preventing integer division

```
int books = 30, months = 7;
double booksPerMonth;
booksPerMonth = static_cast<double>(books) / months;
```

- What about this statement?

```
booksPerMonth = static_cast<double>(books / months);
```

- Displaying a char from its ASCII value

```
int number = 65;
cout << static_cast<char>(number)
```

9/20/06

CS150 Introduction to Computer Science 1

14

Examples of Combined Assignments

Operator	Example Usage	Equivalent To
<code>+=</code>	<code>x += 5;</code>	<code>x = x + 5;</code>
<code>-=</code>	<code>y -= 2;</code>	<code>y = y - 2;</code>
<code>*=</code>	<code>z *= 10;</code>	<code>z = z * 10;</code>
<code>/=</code>	<code>a /= b;</code>	<code>a = a / b;</code>
<code>%=</code>	<code>c %= 3;</code>	<code>c = c % 3;</code>

9/20/06

CS150 Introduction to Computer Science 1

15

Relational Operators, Explained!

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- o All are binary operators
- o Left to right associativity

9/20/06 CS150 Introduction to Computer Science 1 16

Relational Expression

- An expression is a statement that has **value**
- Relational expression: an expression that uses a Relational Operator
 - o its value is a **Boolean value** (True or False)

```
int x = 9, y = 42;
```

```
x > y
```

```
y == x // y = x; is the assignment operator
```

```
x <= (x * y + 99)
```

9/20/06 CS150 Introduction to Computer Science 1 17

Precedence (page 1125)

Precedence Operators (Highest to Lowest)	
(unary negation) -	
* / %	
Arithmetic Operators	
+ -	
> >= < <=	
Relational Operators	
== !=	
= += -= *= /= %=	
Assignment Operators	

9/20/06 CS150 Introduction to Computer Science 1 18

Grouping!

- To override precedence we use grouping symbols, ()
 - `average = (a + b + c) / 3;`
- $(3 + 12) * 2 - 3$
- $4 + 17 \% (3 + 9)$
- $6 - 2 * 9 / ((3 * 4) - 9)$
 - Work from the inside () outward

9/20/06

CS150 Introduction to Computer Science 1

19

Boolean value (True or False)

- How does the computer represent True and False?
- New data type: `bool`

```
bool tValue = true; // 1
bool fValue = false; // 0
```

9/20/06

CS150 Introduction to Computer Science 1

20

Formally defined

```
if( expression )
{
    statement 1;
    statement 2;
    . . .
    statement n;
}
```

Just like a **function**, start at the top and execute in order to the bottom

- What is an expression?

9/20/06

CS150 Introduction to Computer Science 1

21
