

Από C σε C++

- 15.1 Εισαγωγή
- 15.2 C++
- 15.3 Προσθήκη 2 ακεραίων
- 15.4 C++ Standard Library
- 15.5 Header Files

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



2

1.7 History of C and C++

- History of C
 - Evolved from two other programming languages
 - BCPL and B
 - “Typeless” languages
 - Dennis Ritchie (Bell Laboratories)
 - Added data typing, other features
 - Development language of UNIX
 - Hardware independent
 - Portable programs
 - 1989: ANSI standard
 - 1990: ANSI and ISO standard published
 - ANSI/ISO 9899: 1990
 - C99, C11

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



Αντικειμενοστρεφής Προγραμματισμός

Ιστορία:

1967: **Simula67** (Νορβηγία) -> πρώτη αντικειμενοστρεφής γλώσσα

'70: **Smalltalk** (Palo Alto, CA) -> κάθε στοιχείο ένα αντικείμενο

Αρχές '80: ο αντικειμενοστρεφής τρόπος σκέψης εισάγεται σε ακαδημαϊκούς κύκλους

'80: **C++** (Stroustrup, AT&T): σοβαρή, αποδοτική γλώσσα, πρότυπο στη βιομηχανία

1995: **Java** (Sun Microsystems)

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



Αντικειμενοστρεφής Προγραμματισμός

Αποδοχή:

Υψηλότετη τα τελευταία χρόνια

Λόγοι:

- Επιτυχής επίλυση προβλημάτων μικρής όσο και μεγάλης κλίμακας (scalability)
- Προσομοιάζει τις τεχνικές καθημερινών προβλημάτων
- Πολλές διαθέσιμες γλώσσες
- Πολλές διαθέσιμες βιβλιοθήκες
- Η λανθασμένη αντίληψη ότι αν κάποιος γνωρίζει C, μπορεί εύκολα να μάθει C++

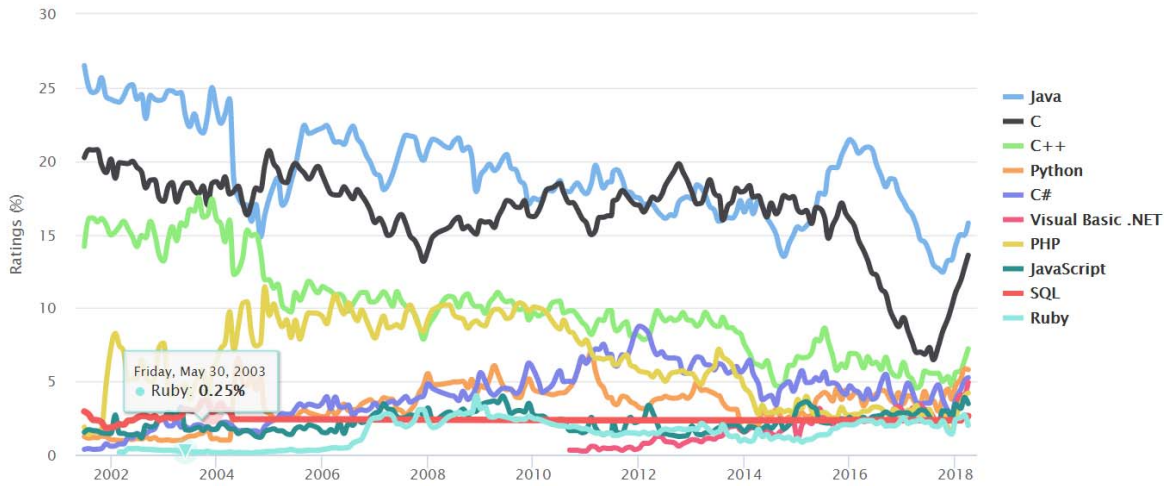
© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



1	1		Java	15.777%
2	2		C	13.589%
3	3		C++	7.218%
4	5	↑	Python	5.803%
5	4	↓	C#	5.265%
6	7	↑	Visual Basic .NET	4.947%
7	6	↓	PHP	4.218%
8	8		JavaScript	3.492%
9	-	↑↑	SQL	2.650%
10	11	↑	Ruby	2.018%

TIOBE Programming Community Index

Source: www.tiobe.com



Repositories	111K
Code	
Commits	
Issues	80M
Topics	332K
Wikis	
Users	29M

Languages	
JavaScript	482,691
Python	274,865
Java	234,201
PHP	154,633
Ruby	130,716
C++	102,961
C	102,961
C#	81,769
Shell	76,072
HTML	75,360

[Advanced search](#) [Cheat sheet](#)

111,477 repository results

Sort: Most forks ▼

tensorflow/tensorflow

C++

★ 96.1k

Computation using data flow graphs for scalable machine learning

tensorflow python machine-learning

Apache-2.0 license Updated 2 hours ago

bitcoin/bitcoin

C++

★ 30.5k

Bitcoin Core integration/staging tree

bitcoin c-plus-plus cryptography p2p

MIT license Updated an hour ago 3 issues need help

opencv/opencv

C++

★ 23.7k

Open Source Computer Vision Library

c-plus-plus opencv computer-vision

Updated 2 hours ago 6 issues need help

BVLC/caffe

C++

★ 23.7k

Caffe: a fast open framework for deep learning.

Updated a day ago

15.2 C++

- C++
 - Βελτιώνει χαρακτηριστικά της C
 - Έχει αντικειμενοστραφείς δυνατότητες
 - Αύξηση ποιότητας λογισμικού και επαναχρησιμοποίησης
 - Αναπτύχθηκε από Bjarne Stroustrup at Bell Labs ('80s)
 - Ονομάστηκε "C with classes"
 - C++ (increment operator) – επαυξημένη έκδοση της C
 - Υπερσύνολο της C
 - μπορούμε να χρησιμοποιήσουμε ένα C++ compiler για να κάνουμε compile C προγράμματα
 - Βαθμιαία επέκτεινε τα C προγράμματα σε C++
- ANSI/ISO C++
- 1998 ISO/IEC 14882:1998 C++98
- 2003 ISO/IEC 14882:2003 C++03
- 2007 ISO/IEC TR 19768:2007 C++TR1
- 2011 ISO/IEC 14882:2011 C++11
- 2014 ISO/IEC 14882:2014 C++14
- 2017 ISO/IEC 14882:2017 C++17



© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



Compilers

- CYGWIN (G++)
- Devc++
- Code::blocks (MinGW)
- GCC
- CLANG
- Microsoft Visual C++ (MSVC)

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



Λόγοι για εκμάθηση C++

- Παιδαγωγικοί
- Απευθείας πρόσβαση στη μνήμη (stack/heap)
- ...και γενικά στο υλικό
- Ταχύτητα!



C++ vs Java

- Απλοϊκά προγράμματα σε Java vs C++
 - Jit compiler optimizations (small chunks allocation, variable scopes...)
 - Some ONLY possible at runtime
 - Better memory management
 - Greedy optimization
 - Processor specific instructions (e.g. SSE, AVX, AVX512, etc.) (x86, ARM, etc.)
- **Carefully** written code
- However: Java → greater performance
 - Less chances to use hardware features
 - No direct stack access



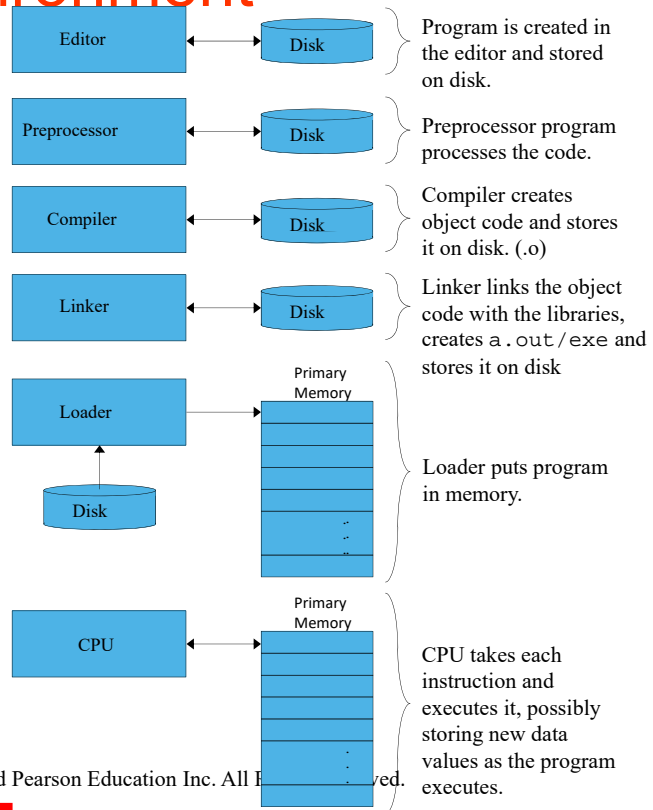
You don't pay for what you don't use (Stroustrup)



1.14 Basics of a Typical C++ Environment

Phases of C++ Programs:

1. Edit
2. Preprocess
3. Compile
4. Link
5. Load
6. Execute



© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



1.14 Basics of a Typical C++ Environment

- Input/output
 - **cin**
 - Standard input stream
 - Normally keyboard
 - **cout**
 - Standard output stream
 - Normally computer screen
 - **cerr**
 - Standard error stream
 - Display error messages

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



1.21 A Simple Program: Printing a Line of Text

- Comments
 - Document programs
 - Improve program readability
 - Ignored by compiler
 - Single-line comment
 - Begin with //
- Preprocessor directives
 - Processed by preprocessor before compiling
 - Begin with #

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



```

1 // Fig. 1.2: fig01_02.cpp
2 // A first program
3 #include <iostream>
4
5 // function main body
6 int main()
7 {
8     std::cout << "Welcome to
9     return 0; //
11
12 } // end function main

```

14

Outline

fig01_02.cpp
output (1 of 1)

Single-line comments.

Function main returns an integer value.

Left brace { begins function body.

Right brace } ends function body.

Stream insertion operator.

Keyword return is one of several means to exit function; value 0 indicates program terminated successfully.

Statements end with a semicolon ;.

every C++ program.

Name of standard library namespace: std.

Welcome to C++!

1.21 A Simple Program: Printing a Line of Text

- Standard output stream object
 - `std::cout`
 - “Connected” to screen
 - `<<`
 - Stream insertion operator
 - Value to right (right operand) inserted into output stream
- Namespace
 - `std::` specifies using name that belongs to “namespace”
`std`
 - `std::` removed through use of `using` statements
- Escape characters
 - `\`
 - Indicates “special” character output

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



1.21 A Simple Program: Printing a Line of Text

Escape Sequence	Description
<code>\n</code>	Newline. Position the screen cursor to the beginning of the next line.
<code>\t</code>	Horizontal tab. Move the screen cursor to the next tab stop.
<code>\r</code>	Carriage return. Position the screen cursor to the beginning of the current line; do not advance to the next line.
<code>\a</code>	Alert. Sound the system bell.
<code>\\</code>	Backslash. Used to print a backslash character.
<code>\"</code>	Double quote. Used to print a double quote character.

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.




```

1 // Fig. 1.4: fig01_04.cpp
2 // Printing a line with multiple statements.
3 #include <iostream>
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "Welcome ";
9     std::cout << "to C++!\n";
10
11     return 0; // indicate that program ended successfully
12
13 } // end function main

```

```
Welcome to C++!
```

Multiple stream insertion statements produce one line of output.



fig01_04.cpp
(1 of 1)

fig01_04.cpp
output (1 of 1)

© 2003 Prentice Hall, Inc.
All rights reserved.

```

1 // Fig. 1.5: fig01_05.cpp
2 // Printing multiple lines with a single statement
3 #include <iostream>
4
5 // function main begins program execution
6 int main()
7 {
8     std::cout << "Welcome\n\tto\n\nC++!\n";
9
10     return 0; // indicate that program ended successfully
11
12 } // end function main

```

```
Welcome
to

C++!
```

Using newline characters to print on multiple lines.



fig01_05.cpp
(1 of 1)

fig01_05.cpp
output (1 of 1)

© 2003 Prentice Hall, Inc.
All rights reserved.

15.3 A Simple Program: Adding Two Integers

- Επεκτάσεις αρχείων
 - C files: .c
 - C++ files: .cpp, .cxx, .C (uppercase)
- Διαφορές
 - `<i ostream>` - input/output stream header file
 - Επιστρεφόμενοι τύποι – όλες οι συναρτήσεις πρέπει να δηλώνουν τον τύπο επιστροφής
 - C δεν το απαιτεί, αλλά η C++ το απαιτεί
 - Κενή λίστα παραμέτρων =καμία παράμετρος
 - Οι μεταβλητές C++ μπορεί να οριστούν σχεδόν οπουδήποτε, ενώ της C, πριν τις εκτελέσιμες εντολές

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



15.3 A Simple Program: Adding Two Integers (II)

- Είσοδος/Εξοδος σε C++
 - πραγματοποιείται με streams χαρακτήρων
 - Streams στέλνονται σε αντικείμενα εισόδου/εξόδου
- Έξοδος
 - `std::cout` – πρότυπο stream εξόδου (συνδέεται με οθόνη)
 - `<< stream` τελεστής ένθεσης ("put to")
 - `std::cout << "hi";`
 - τοποθετεί "hi" στο `std::cout`, που εκτυπώνεται στην οθόνη
- Είσοδος
 - `std::cin` – πρότυπο αντικείμενο εισόδου (συνδέεται με πληκτρολόγιο)
 - `>> stream` τελεστής εξαγωγής ("get from")
 - `std::cin >> myVariable;`
 - Παίρνει το stream από το πληκτρολόγιο και το τοποθετεί στο `myVariable`

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



15.3 A Simple Program: Adding Two Integers (III)

- `std::endl`
 - "end line"
 - Stream manipulator – εκτύπωση νέας γραμμής και flushes τον buffer εξόδου
 - Μερικά συστήματα δεν επιδεικνύουν την έξοδο μέχρι να υπάρχουν αρκετοί χαρακτήρες
 - `std::endl` αναγκάζει το κείμενο να εκτυπωθεί
- `using` εντολές
 - Επιτρέπουν απομάκρυνση του `std::` prefix
 - Συζητείται αργότερα
- Cascading
 - Can have multiple `<<` or `>>` operators in a single statement

```
std::cout << "Hello " << "there" << std::endl;
```

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



```
1 // Fig. 15.1: fig15_01.cpp
2 // Addition program
3 #include <iostream>
4
5 int main()
6 {
7     int integer1;
8
9     std::cout << "Enter first integer\n";
10    std::cin >> integer1;
11
12    int integer2, sum;           // declaration
13
14    std::cout << "Enter second integer\n";
15    std::cin >> integer2;
16    sum = integer1 + integer2;
17    std::cout << "Sum is " << sum << std::endl;
18
19    return 0; // indicate that program ended successfully
20 } // end function main
```

```
Enter first integer
45
Enter second integer
72
Sum is 117
```



Outline

fig15_01.cpp

15.4 C++ Standard Library

- C++ προγράμματα χτίζονται από
 - Συναρτήσεις
 - Κλάσεις
 - Οι περισσότεροι προγραμματιστές χρησιμοποιούν library functions
- Δύο τρόποι εκμάθησης C++
 - Εκμάθηση γλώσσας
 - Εκμάθηση library functions
- Δημιουργία δικών σας συναρτήσεων
 - Πλεονέκτημα: γνώση ακριβώς πως δουλεύουν
 - Μειονέκτημα: χρονοβόρα, δύσκολο να διατηρηθεί η αποτελεσματικότητα και η καλή σχεδίαση

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



15.5 Header Files

- Header αρχεία
 - Κάθε standard library έχει header files
 - περιλαμβάνει function prototypes, data type definitions, και constants
 - Files ending with . h are "old-style" headers
- User defined header αρχεία
 - Create your own header file
 - End it with . h
 - Use `#include "myFile.h"` in other files to load your header

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



15.5 Header Files

Standard library header file	Explanation
<cassert>	Contains macros and information for adding diagnostics that aid program debugging. The old version of this header file is <assert.h>.
<cctype>	Contains function prototypes for functions that test characters for certain properties, that can be used to convert lowercase letters to uppercase letters and vice versa. This header file replaces header file <ctype.h>.
<cmath>	Contains the floating-point size limits of the system. This header file replaces header file <float.h>.
<climits>	Contains the integral size limits of the system. This header file replaces header file <limits.h>.
<cmath>	Contains function prototypes for math library functions. This header file replaces header file <math.h>.
<cstdio>	Contains function prototypes for the standard input/output library functions and information used by them. This header file replaces header file <stdio.h>.
<cstdlib>	Contains function prototypes for conversions of numbers to text, text to numbers, memory allocation, random numbers and various other utility functions. This header file replaces header file <stdlib.h>.
<cstring>	Contains function prototypes for C-style string processing functions. This header file replaces header file <string.h>.
<ctime>	Contains function prototypes and types for manipulating the time and date. This header file replaces header file <time.h>.
<iostream>	Contains function prototypes for the standard input and standard output functions. This header file replaces header file <iostream.h>.

Fig. 15.2 Standard library header files. (Part 1 of 3)

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



15.5 Header Files

Standard library header file	Explanation
<iomanip>	Contains function prototypes for the stream manipulators that enable formatting of streams of data. This header file replaces <iomanip.h>.
<fstream>	Contains function prototypes for functions that perform input from files on disk and output to files on disk. This header file replaces header file <fstream.h>.
<utility>	Contains classes and functions that are used by many standard library header files.
<vector>, <list>, <deque>, <queue>, <stack>, <map>, <set>, <bitset>	These header files contain classes that implement the standard library containers. Containers are used to store data during a program's execution.
<functional>	Contains classes and functions used by standard library algorithms.
<memory>	Contains classes and functions used by the standard library to allocate memory to the standard library containers.
<iterator>	Contains classes for accessing data in the standard library containers.
<algorithm>	Contains functions for manipulating data in standard library containers.
<exception>, <stdexcept>	These header files contain classes that are used for exception handling (discussed in Chapter 23).

Fig. 15.2 Standard library header files. (Part 2 of 3)

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



15.5 Header Files

Standard library header file	Explanation
<string>	Contains the definition of class <code>string</code> from the standard library.
<sstream>	Contains prototypes for functions that perform input from strings in memory and output to strings in memory.
<locale>	Contains classes and functions normally used by stream processing to process data in the natural form for different languages (e.g., monetary formats, sorting strings, character presentation, etc.).
<limits>	Contains classes for defining the numerical data type limits on each computer platform.
<typeinfo>	Contains classes for run-time type identification (determining data types at execution time).

Fig. 15.2 Standard library header files. (Part 3 of 3)

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



28

1.23 Memory Concepts

```
std::cin >> integer1;
```

– Assume user entered 45

integer1	45
----------	----

```
std::cin >> integer2;
```

– Assume user entered 72

integer1	45
----------	----

integer2	72
----------	----

```
sum = integer1 + integer2;
```

integer1	45
----------	----

integer2	72
----------	----

sum	117
-----	-----

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



1.24 Arithmetic

- Arithmetic calculations
 - *
 - Multiplication
 - /
 - Division
 - Integer division truncates remainder
 - $7 / 5$ evaluates to 1
 - %
 - Modulus operator returns remainder
 - $7 \% 5$ evaluates to 2

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



1.24 Arithmetic

- Rules of operator precedence
 - Operators in parentheses evaluated first
 - Nested/embedded parentheses
 - Operators in innermost pair first
 - Multiplication, division, modulus applied next
 - Operators applied from left to right
 - Addition, subtraction applied last

Operator(s)	Operation(s)	Order of evaluation (precedence)
()	Parentheses	Evaluated first. If the parentheses are nested, the expression in the innermost pair is evaluated first. If there are several pairs of parentheses “on the same level” (i.e., not nested), they are evaluated left to right.
*, /, or %	Multiplication Division Modulus	Evaluated second. If there are several, they are evaluated left to right.
+ or -	Addition Subtraction	Evaluated last. If there are several, they are evaluated left to right.

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



1.25 Decision Making: Equality and Relational Operators

- **if** structure
 - Make decision based on truth or falsity of condition
 - If condition met, body executed
 - Else, body not executed
- Equality and relational operators
 - Equality operators
 - Same level of precedence
 - Relational operators
 - Same level of precedence
 - Associate left to right

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



1.25 Decision Making: Equality and Relational Operators

Standard algebraic equality operator or relational operator	C++ equality or relational operator	Example of C++ condition	Meaning of C++ condition
<i>Relational operators</i>			
>	>	$x > y$	x is greater than y
<	<	$x < y$	x is less than y
\geq	\geq	$x \geq y$	x is greater than or equal to y
\leq	\leq	$x \leq y$	x is less than or equal to y
<i>Equality operators</i>			
=	==	$x == y$	x is equal to y
\neq	!=	$x != y$	x is not equal to y

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



1.25 Decision Making: Equality and Relational Operators

- **using** statements
 - Eliminate use of **std::** prefix
 - Write **cout** instead of **std::cout**

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



```

1 // Fig. 1.14: fig01_14.cpp
2 // Using if statements, relational
3 // operators, and equality operators.
4 #include <iostream>
5
6 using std::cout; // program uses cout
7 using std::cin;  // program uses cin
8 using std::endl; // program uses endl
9
10 // function main begins program
11 int main()
12 {
13     int num1; // first number
14     int num2; // second number
15
16     cout << "Enter two integers
17     << "the relationships between
18     cin >> num1 >> num2; // read
19
20     if ( num1 == num2 )
21         cout << num1 << " is equal to " << num2 << endl;
22
23     if ( num1 != num2 )
24         cout << num1 << " is not equal to " << num2 << endl;
25

```

using statements eliminate need for std:: prefix.

Declare variables.

Can write cout and cin without std:: prefix.

if structure compares values of num1 and num2 to test if

If condition is true (i.e., values are equal), execute this statement.

if structure compares values of num1 and num2 to test if

If condition is true (i.e., values are not equal), execute this statement.



fig01_14.cpp
(1 of 2)

34

```

26  if ( num1 < num2 )
27      cout << num1 << " is less than " << num2 << endl;
28
29  if ( num1 > num2 )
30      cout << num1 << " is greater than " << num2 << endl;
31
32  if ( num1 <= num2 )
33      cout << num1 << " is less than or equal to "
34          << num2 << endl;
35
36  if ( num1 >= num2 )
37      cout << num1 << " is greater than or equal to "
38          << num2 << endl;
39
40  return 0;  // indicate that program ended successfully
41
42 } // end function main

```

```

Enter two integers, and I will tell you
the relationships they satisfy: 22 12
22 is not equal to 12
22 is greater than 12
22 is greater than or equal to 12

```



Outline

fig01_14.cpp

Statements may be split over several lines.

fig01_14.cpp
output (1 of 2)

© 2003 Prentice Hall, Inc.
All rights reserved.

```

Enter two integers, and I will tell you
the relationships they satisfy: 7 7
7 is equal to 7
7 is less than or equal to 7
7 is greater than or equal to 7

```



Outline

fig01_14.cpp
output (2 of 2)

© 2003 Prentice Hall, Inc.
All rights reserved.