Introduction to Programming with C++

Chieh-Sen (Jason) Huang

Department of Applied Mathematics

National Sun Yat-sen University

INTRODUCTION TO
PROGRAMMING
WITH
C++

Third Edition

Contents are based on book by Y. Daniel Liang

Selections

- The program can decide which statements to execute based on a condition.
- Like all high-level programming languages, C++ provides selection statements: statements that let you choose actions with alternative courses.

```
if (radius < 0)
{
  cout << "Incorrect input" << endl;
}
else
{
  area = radius * radius * PI;
  cout << "The area for the circle of radius " << radius
     << " is " << area << endl;
}
```

- Selection statements use conditions that are Boolean expressions.
- A Boolean expression is an expression that evaluates to a Boolean value: true or false.

The bool Data Type

- The bool data type declares a variable with the value either true or false.

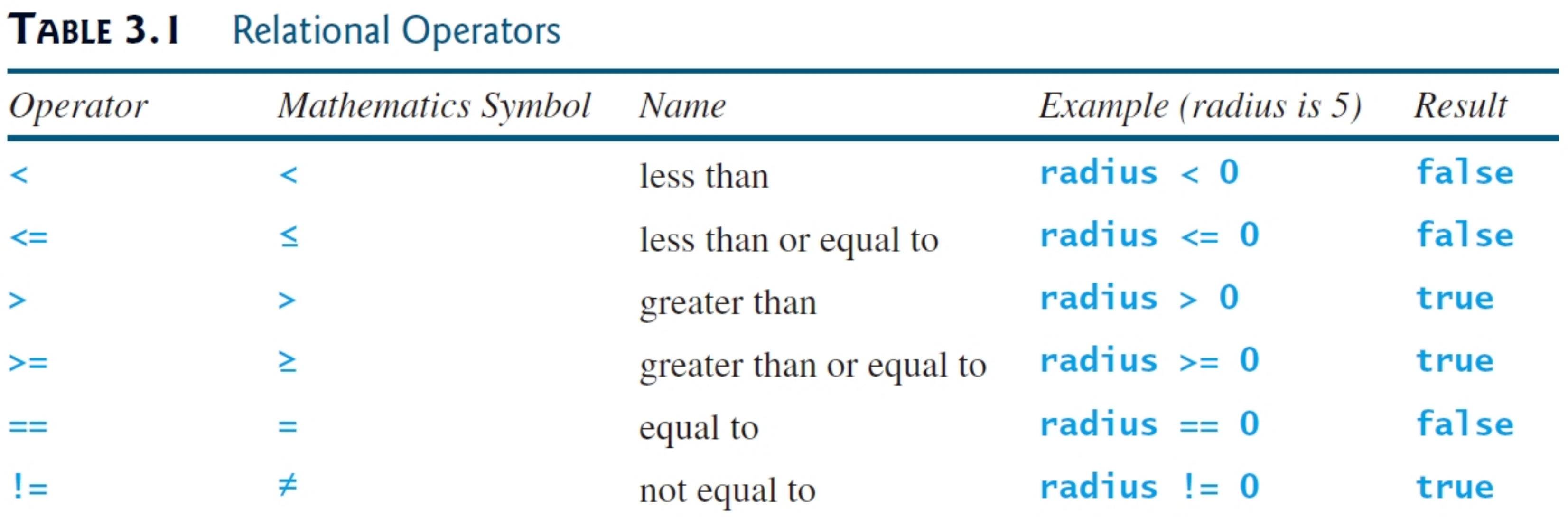


TABLE 3.1 Relational Operators

Operator	*Mathematics Symbol*	*Name*	*Example (radius is 5)*	*Result*
`<`	$<$	less than	`radius < 0`	`false`
`<=`	$\leq$	less than or equal to	`radius <= 0`	`false`
`>`	$>$	greater than	`radius > 0`	`true`
`>=`	$\geq$	greater than or equal to	`radius >= 0`	`true`
`==`	$=$	equal to	`radius == 0`	`false`
`!=`	$\neq$	not equal to	`radius != 0`	`true`

- A variable that holds a Boolean value is known as a Boolean variable.

```
bool lightsOn = true;
```

- True and False are Boolean literals. They are keywords and cannot be used as identifiers in your program.

The bool Data Type

- Internally, C++ uses 1 to represent true and 0 for false. If you display a bool value to the console, 1 is displayed if the value is true and 0 if it is false.

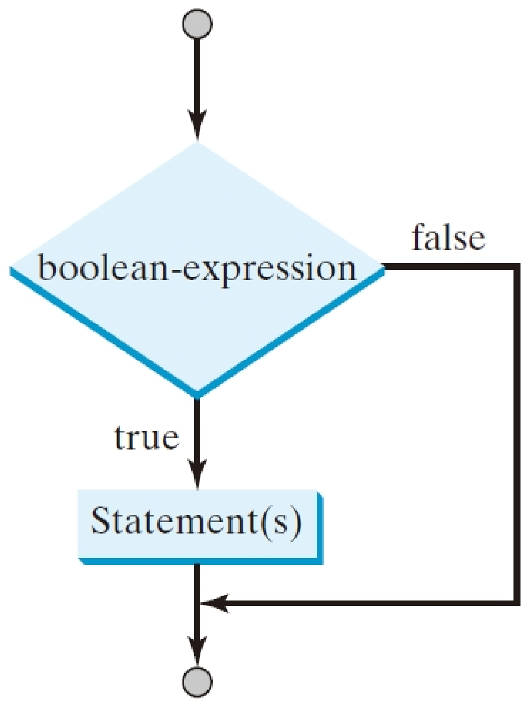
  In-Class Exercise: 3.3

if Statements

- An **if** statement is a construct that enables a program to specify alternative path of execution.
- A one-way if statement executes an action if and only if the condition is true.

```
if (boolean-expression)
{
statement(s);
}
```

A flowchart is a diagram that describes an algorithm or process.



SimpleIfDemo.cpp

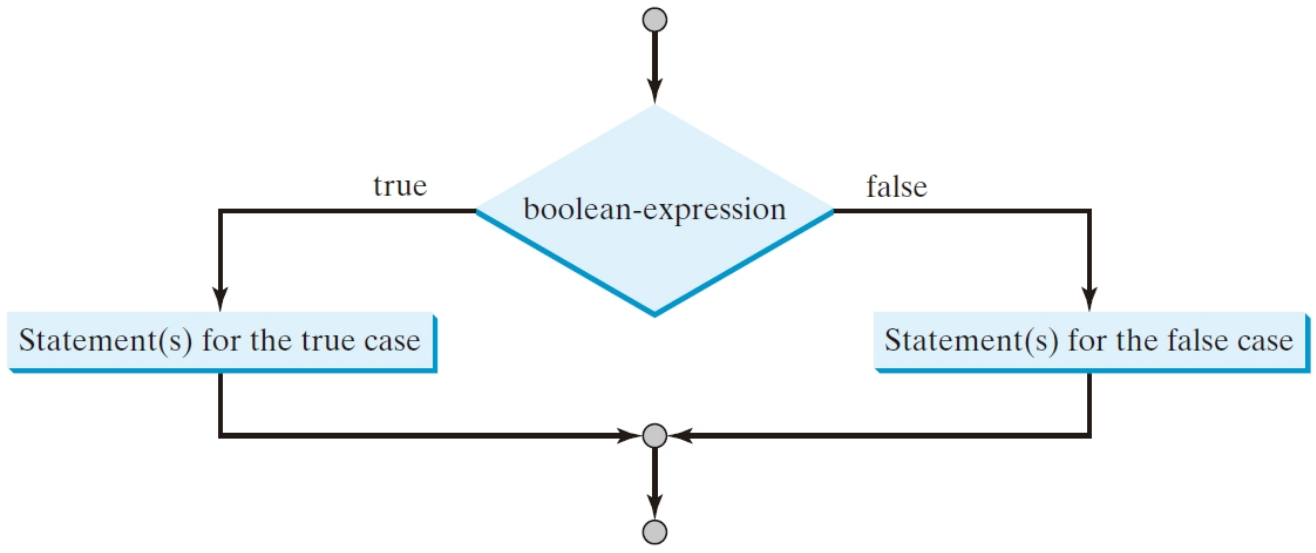
```
int main()
{
  int number; // List3_1.cpp
  // Prompt the user to enter an integer
  cout << "Enter an integer: ";
  cin >> number;

  if (number % 5 == 0)
   cout << "HiFive" << endl;

  if (number % 2 == 0)
   cout << "HiEven" << endl;

   return 0;
}
```

```
Enter a radius: 4 enter
HiEven
```

Two-Way if-else Statements

- An if-else statement decides which statements to execute based on whether the condition is true or false.

```
if (boolean-expression)
{
  statement(s)-for-the-true-case;
}
else
{
  statement(s)-for-the-false-case;
}
```


true
boolean-expression
false
Statement(s) for the true case
Statement(s) for the false case

Two-Way if-else Statements

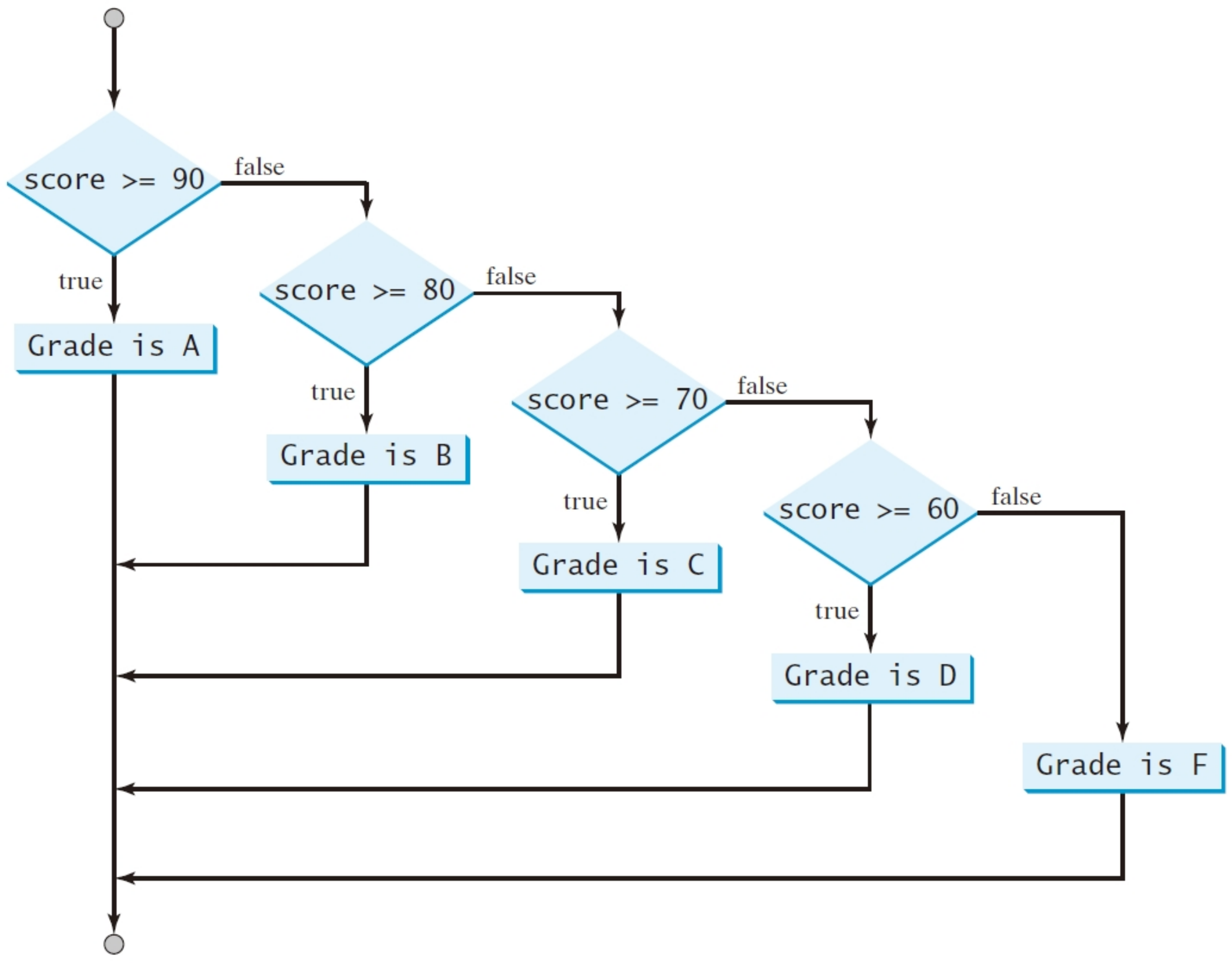
```
if (number % 2 == 0)
  cout << number << " is even.";
else
  cout << number << " is odd.";
```

In-Class Exercise: Write a code to check whether a number is even or odd.

- An if statement can be inside another if statement to form a nested if statement.

```
if (score >= 90.0)
  cout << "Grade is A";
else if (score >= 80.0)
  cout << "Grade is B";
else if (score >= 70.0)
  cout << "Grade is C";
else if (score >= 60.0)
  cout << "Grade is D";
else
  cout << "Grade is F";
```

A multi-way if-else statement



Homework 3.1: Write a code to decide if a year is a leap year or not.

Logical Operators

The logical operators !, &&, and || can be used to create a compound Boolean expression.
(& ampersand, ! exclamation mark, | vertical bar, pipe).

Operator	Name	Description
!	not	logical negation
&&	and	logical conjunction
||	or	logical disjunction

```
if (number % 2 == 0 && number % 3 == 0)
  cout << number << " is divisible by 2 and 3." << endl;

if (number % 2 == 0 || number % 3 == 0)
  cout << number << " is divisible by 2 or 3." << endl;

if ((number % 2 == 0 || number % 3 == 0) &&
     !(number % 2 == 0 && number % 3 == 0))
  cout << number << " divisible by 2 or 3, but not both." << endl;
```

In-Class Exercise: Write a code to decide the number of days for a given month number. Anwer 28 or 29 if 2 (February) is entered.

Homework 3.2: Check-point 3.23 (page 113). Redo the leap year hw using logical operators.

Switch Statements

- A switch statement executes statements based on the value of a variable or an expression.
- The switch-expression must yield an integral value and always be enclosed in parentheses.

```
switch (month)
{
  case 2: cout << "month number" << month<< "is 28 or 29 days."  << endl;
          break;
  case 4:
  case 6:
  case 9:
  case 11: cout << "month number" << month << "is 30 days." << endl;
           break;
  case 1:
  case 3:
  case 5:
  case 7:
  case 8:
  case 10:
  case 12: cout << "month number" << month << "is 31 days."<<endl;
           break;
  default: cout << "Error: invalid month number" << endl;
}
```

Conditional Expressions

- A conditional expression evaluates an expression based on a condition.
- Conditional expressions have a completely different structure and do not include an explicit if.

```
boolean-expression ? expression1 : expression2;
--------------------------------------------------------------
y = x > 0 ? 1 : -1;
--------------------------------------------------------------
if (x > 0)
  y = 1;
else
  y = -1;
--------------------------------------------------------------
cout << (num % 2 == 0 ? "num is even" : "num is odd") << endl;
```

In-Class Exercise: Check-point 3.35 (page 122).

Operator Precedence and Associativity

- Use parentheses to force an evaluation order.

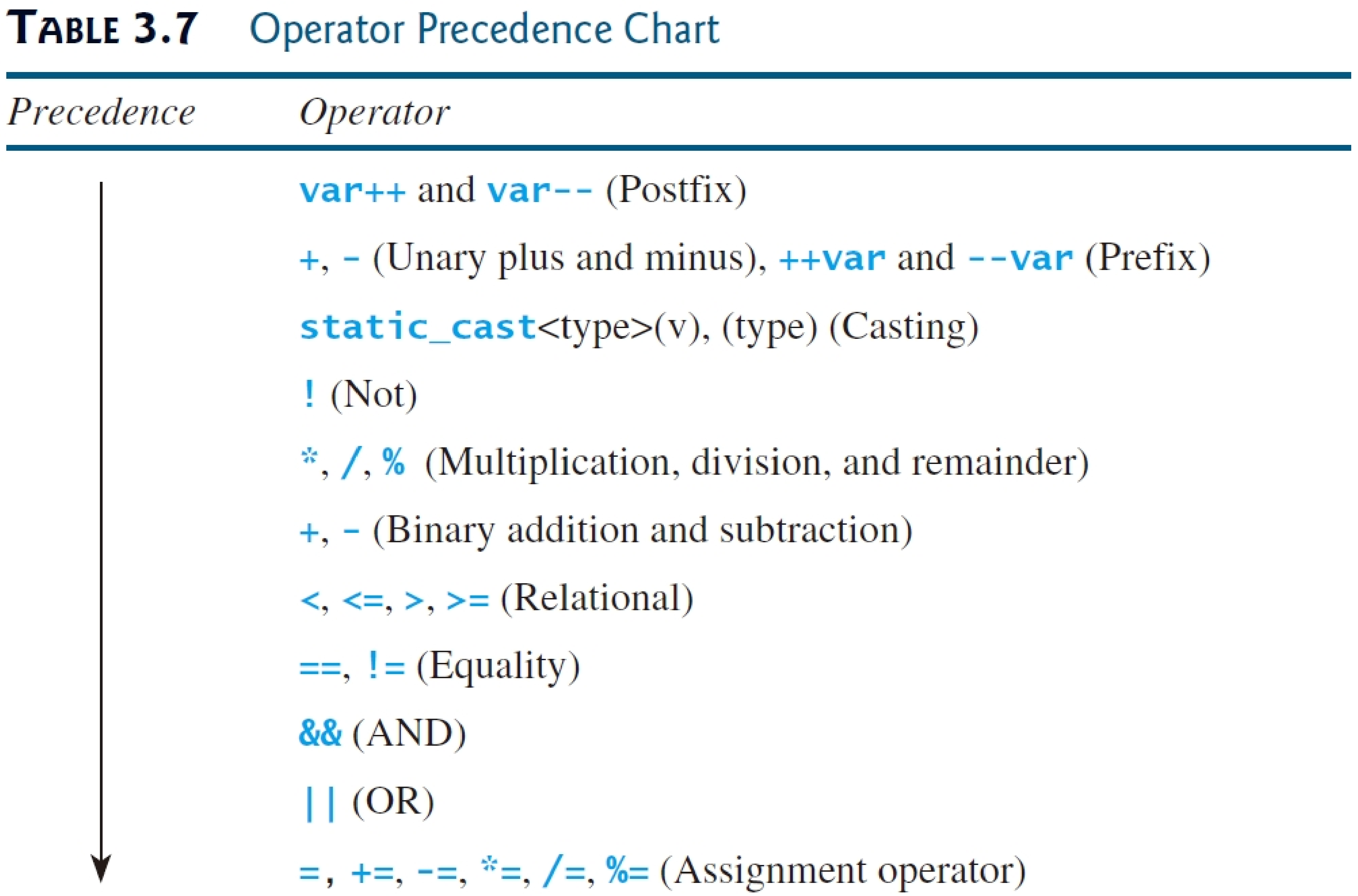


TABLE 3.7 Operator Precedence Chart

Precedence	*Operator*
	`var++` and `var--` (Postfix)
	`+`, `-` (Unary plus and minus), `++var` and `--var` (Prefix)
	`static_cast`<type>(v), (type) (Casting)
	`!` (Not)
	`*`, `/`, `%` (Multiplication, division, and remainder)
	`+`, `-` (Binary addition and subtraction)
	`<`, `<=`, `>`, `>=` (Relational)
	`==`, `!=` (Equality)
	`&&` (AND)
	`||` (OR)
	`=`, `+=`, `-=`, `*=`, `/=`, `%=` (Assignment operator)

Homework 3.3: Programming exercise, 3.1.