CS150 Intro to CS I

Fall 2012

Chapter 4
Making Decisions

- Reading: Chapter 4 (4.1 pp. 149-154; 4.2 pp. 154-162; 4.3 pp. 162-165)

- Good Problems to Work: pp.153-154[4.1, 4.2]; p. 162[4.5, 4.7, 4.9], p.165[4.10, 4.11]

Conditionals

- So far, we can Input, Output and Calculate

- How can we explore relationships between data?

- How can our program only do certain things sometimes?

Decisions

- Relational Expressions allow our program to *make a decision*

  - ➤ based on the **data** in the program

- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a Relational Operator

  - Relational expressions produce a value that is a Boolean value (True or False)

```
int height = 32;
int minHeight = 34;
bool bVal;

bVal = height > minHeight;

bVal = height == minHeight;

bVal = height >= (minHeight - 2);
```

Relational Operators

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

- All operators are binary

- Each operator is left associative. What does this mean?

Operator Precedence (highest to lowest)

Unary plus & minus	+ -	Left associative
Multiplication, division, and modulus	* / %	Left associative
Addition & subtraction	+ -	Left associative
Relational operators	< <= > >=	Left associative
Relational operators	== !=	Left associative
Assignment	=	Right associative

const Declarations

- Constant declaration

```
const double PI = 3.14;

const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed

- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

```
int width = 99;
const int HEIGHT = 42;
bool bVal;

bVal = width > HEIGHT;

bVal = -width <= HEIGHT;

bVal = width != HEIGHT;

bVal = width == (width + 1);

bVal = width == width + 1;
```

Relational Operators work on Integers, Floating point numbers, and Characters.

The if Statement

- We execute each statement in our program in order

- What `if` we only want to execute a statement sometimes?

- The `if` statement

- Single-alternative `if`

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

Practice

```cpp
int age;
const int VOTING_AGE = 18;

cout << "Enter your age:";
cin >> age;

if ( VOTING_AGE <= age )
{
  cout << age << " >= " << VOTING_AGE;
  cout << " You can vote!" << endl;
}
```

What values of age need to be entered to fully test this code?

Practice

Write a C++ program segment that allows the user the ability to enter 3 exam scores one at a time. If the average exam score is higher than 60, output "You Passed"; otherwise output "You Failed". Sum the scores as they are entered.

1. What variables (including their types) are needed in this program segment?

Hint: If the variable count contains the value 0, what is the value of count after executing the statement count = count + 1;

2. What do you need to do conditionally?

3. What data is necessary to use in the condition?

Problem

Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders.

1. If the order is $30 or less then shipping will cost $5

2. If the order is over $30 then shipping will be $3.

Write a C++ program to solve this problem

Problem Modification

The bookstore has now changed it's shipping policy so that

1. If the order is $30 or less, shipping is $5

2. If the order is over $30 but less than $50, shipping is $3

3. If the order is $50 or more then shipping is $2

What would we need to change in the program?