Announcements

- 1st Assignment due next Monday, Sep 15, 2003
- 1st Exam next Friday, Sep 19, 2003
- 1st Quiz next week
- Start reading Chapter 4

Program

- Write a program to convert a temperature in degrees Fahrenheit to degrees Celsius
  - Celsius = (5/9)*(Fahrenheit - 32)

Program

- Write a program that asks the user for their name and phone number and prints the information on one line.
- How would we add their address to the information read in?

General Programming Style

- Header of comments
- Comment each variable (or similar groups of variables)
- Comment each major step of program
- Include spaces between lines of code when appropriate
- Indent the body of your program (declaration and executable statements)
- We'll learn more formatting rules later

Arithmetic Formatting

- Include spaces around operators and =
- Can include any line breaks in single executable statement
  - Indent any continued arithmetic expression
    - ✓ x = sum_of_prices/total_prices
    - ✓ + std_dev_prices;
    - ✓ (Yes, you can continue an assignment statement on the next line)
  - Do not put a line break in the middle of an identifier, reserved word or string literal
- Do not put spaces between <<, >>, //

Programming Errors (Bugs)

- Syntax error
  - Error with grammar rules
  - Compiler will catch
- Run-time error
  - Program compiles, but does not run to completion
  - Eg.: Division by zero
- Logic error
  - Program completes, but wrong answer
  - Eg.: Calculating area of circle using wrong value for pi

Example Problems

- State order of evaluation of the operators:

```
x = 7 + 3 * 6 / 2 - 1;

x= 2 % 2 + 2 * 2 - 2 / 2;

x = (3 * 9 * (3 + (9 * 3 / (3))));
```

What is output?

x=2 and y = 3

```
cout << x;
cout << x + x;
cout << “x=“;
cout << “x=“ << x;
cout << x + y << “=“ << y + x;
z = x+y;
cin >>x>>y;
//cout << “x + y = “ << x + y;
```

Programs

- Write a program that reads in two numbers and prints their sum, difference, product and quotient

Chapter 4: Selection Structures

- We’re now ready to start learning more complicated executable statements
- Up until now, all the executable statements execute in our program
- Selection structures (if and switch) allow us to conditionally execute statements
- Read Sections 4.1 - 4.3

Examples

- If someone enters an illegal value for input, we can print out an error message
- We can give the user choices for what they want the program to do

Control Structures

- We have seen sequential flow of a program
- Now we will learn control structures for selection
- Next we will learn control structures for repetition

Logical Expressions

- The primary selection statement is **`if`**
- The format is
  - `if (some condition is true)`
  - `  execute some statement(s)`
- How do we express the condition?
- Example

```
if (weight > 100.0)
    shipCost = 10.00;
else
    shipCost = 5.00;
```

Relational & Equality Operators

<	less than	relational
>	greater than	relational
<=	less than or equal to	relational
>=	greater than or equal to	relational
==	equal to	equality
!=	equal to	equality

Logical expressions

- Have value true (1) or false (0)
- Include relational or equality operators
- When used in an if statement, called a condition
- Examples:
  - `X <= 0`
  - `power < maxPower`
  - `x >=y`
  - `item > minItem`
  - `momOrDad == 'M'`
  - `num != sentinel`

Logical Operators

- Operators combine logical expressions
- Operators:

&&	and
||	or
!	not

- Examples:
  - `(salary < minSalary) || (dependents > 5)`
  - `(temp > 90.0) && (humidity > 0.9)`

Evaluating Expressions Or

- `(expr1 || expr2)`
  - Either expr1 or expr2 is true
- Examples:
  - `(salary < minSalary) || (dependents > 5)`
  - To qualify for financial aid, salary has to be less than some minimum salary or the number of dependents is greater than 5
  - Only one condition has to be true

Evaluating Expressions And

- `(expr1) && (expr2)`
- Both expr1 and expr2 have to be true
- Example:

```
(temp > 90.0) && (humidity > 0.9)
```

These are unbearable heat and humidity conditions

Both must be true for the entire expression to be true

Not

- Logical complement or negation: !
- Unary operator
- Examples:

```
!((salary < minSalary) && (dependents > 5))
```

What makes this true? False?

```
winningRecord && !probation
```

How do we evaluate?

- `x < min + max`
- `min <= x && x <= max`
- `!x == y`

Operator Precedence

`!, +, -`	Negation, Unary +, -
`*,/,%`	Mult, div, mod
`+, -`	Add, Subtract
`<, <=, >, >=`	Relational
`==, !=`	Equality
`&&`	And
`||`	Or
`=`	Assignment

Examples

- Assume x is 3.0, y is 4.0, z is 2.0 and flag is false
- What is the value of these expressions?

```
!flag
x + y/z <= 3.5
!flag || (y + z >= x - z)
!(flag || (y + z >= x - z)
```