What Actions Do We Have Part 2

Today

- Last time we
  - Examined how data is stored in memory
  - Started talking about Executable statements
    - Assignment statements
    - Output statements
- Today we will
  - Cover input statements
  - Start looking at arithmetic statements

Input

- Input operator (extraction operator): **>>**
- Gets input from some device/file
- Standard input (from keyboard): **cin**
- Whatever the user types in is stored in the variable to the right of the operator (the right operand)
- That variable must have already been declared
  - Given a data type and allocated space in memory
- When reading in the data typed by the user
  - Any spaces before the data item are skipped
  - Continues to read until the user hits return

Input

- Examples:
  ```
  cin >> miles;
  ```
- The variable **miles** must have already been declared
  ```
  int num1;
  int num2;
  cin >> num1 >> num2;
  ```

Problem

- Write the C++ statements necessary to perform the following operations:
  - Display the message below onto the screen
    ```
    "C++      is a useful
    language to know"
    ```
  - Read in from the user their initials (assume there are only two) and their age

Problem

- What is the output?
  ```
  cout << "My name is: ";
  cout << "Doe, Jane." << endl;
  cout << "I live in ";
  cout << "Ann Arbor, MI ";
  cout << "and my zip code is "
     << 48109 << ". " << endl;
  ```

What is the Output?

```
cout << "Enter two numbers: ";
cin >> a >> b;
a = a + 5.0;
b = 3.0 * b;
cout << "a = " << a << endl;
cout << "b = " << b << endl;
```

- Assume 5.0 and 7.0 are entered for a & b

What is the Output?

- Assume x = 2, y = 3

```
cout << x;
cout << x + x;
cout << "x=";
cout << x + y << " = " << y + x;
z = x + y;
cin >> x >> y;
// cout << "x + y = " << x + y;
cout << "\n";
```

Arithmetic Expressions

- Arithmetic expressions manipulate numeric data
- We've already seen simple ones
- The main arithmetic operators are
  - +   addition
  - -   subtraction
  - *   multiplication
  - /   division
  - %   modulus

+, -, and *

- Addition, subtraction, and multiplication behave in C++ in the same way that they behave in algebra

```
int num1, num2, num3, num4, sum, mul;
num1 = 3;
num2 = 5;
num3 = 2;
num4 = 6;
sum = num1 + num2;
mul = num3 * num4;
```

Division /

- The division operator can be used with both integers and doubles
- If the operands are both **doubles**, the result is a **double**
  - Example: **7.0 / 2.0 is 3.5**
- If the operands are both **ints**, the result is an **int**
  - Any fractional part in integer division is discarded
  - Example: **7 / 2 is 3**
- If mixed, the **int** operand is converted to a **double** and the result is a **double**
  - Example: **5 / 2.5 is 2.0**

Division

- Divisor (second operand) cannot be 0
- Division with negative integers may or may not be allowed
  - It depends on the compiler

Summary

- In today's lecture we learnt
  - How to read in input entered by the user using the keyboard
  - How to write complex arithmetic expressions
    - + - * /

- We have covered p. 26 - 33 of your textbook

3