Module 05
Problem Solving with Variables

Intro to Computer Science 1 - C++
Professor Scott Frees

Dialog with the user

It is important to practice working with inputs and presenting outputs.

Soon you will realize that programming is not hard because of *syntax*, its hard because you need to carefully organize your thoughts.

- Flowcharts, diagrams, and thinking are all pre-requisites for good programs!

Programming Example 03

Calculate the cost to ship an item:

- User enters weight
- User enters miles
- Fixed rate of $2.50/pound, per 200 miles
- You output the total cost

Programming Example 04

Ask the user for 2 inputs:

- Base price of an item
- Sales tax rate (let user enter 0.07 for 7%)
- You calculate cost after tax and print it out to the user.
- Don’t worry about decimal places (for now)

Programming Example 05

Ask the user to enter a total number of cents (integer)

Compute how many quarters, dimes, nickels, and pennies should be used to create change.

What about doubles?

Instead of asking the user for integers, lets have them enter "11.56" and instead convert that to 1156 cents

What happens when user enters: 1.13?

    Is the computer wrong?

Remember bits and bytes!

It turns out there are many decimal numbers that cannot be represented perfectly in a fixed-bit notation.

- 1.13 happens to be one of them… its actually stored as 1.129999999999…
- 1.1299999999…x 100 = 112.9999999…
- When stored as an integer, 112.999999 is 112!

Solution: $$\text{amount} = \underbrace{(a * 100\ ) + 0.5;}_{\text{double is cast to int, after 0.5 is added}}$$

Casting and Promotion

- C++ will only perform mathematics on “like” data types.
- To do this, C++ automatically “promotes” to lower “ranking” data type.
- Data types with larger data ranges have higher rank

```
double > float > int > char > bool
```

This is why 5/2 is 2, but 5.0 / 2 is 2.5

Casting and Promotion

Generally it is a bad idea to cast from "bigger" types to smaller types

Data may be lost (1.8 becomes 1)

Sometimes it is needed though (our last example)

There is specific syntax to avoid warnings

```
double y = 5.5;
int x = (int) y;
int x = int (y)
int x = static_cast<int>(y)
```

Recommended