CS150 Intro to CS I

Fall 2015

Chapter 2
Introduction to C++

- Reading: Chapter 2 (2.14 pp.61-69; 3.2 pp.89-99)
- Good Problems to Work: p. 69[2.21, 2.23, 2.24], pp.96-98[3.7, 3.8, 3.10]

Problem

- Write a program that will compute the tip on a restaurant bill for a patron with a $44.50 meal charge. The tip must be 15% of the total bill. Display the meal cost, tip amount, and total bill on the screen.

Assigning `float`s to `int`s

```
int mileage;
mileage = 42.7;
cout << mileage;
```

- What is the output?

Assigning doubles to ints

- What is the output?

```
int intVariable;
double doubleVariable = 78.9;
intVariable = doubleVariable;
cout << intVariable;
```


Arithmetic Expressions

- Arithmetic expressions manipulate numeric data
- The main arithmetic operators are
  - `+` addition
  - `-` subtraction
  - `*` multiplication
  - `/` division
  - `%` modulus (remainder of integer division)

Division

- What is the output?

1. ```
   int average;
   average = 100 / 2;
   cout << average;
   ```

2. ```
   int average;
   average = 100 / 3;
   cout << average;
   ```


Division

- **`average = 100 / 3;`**
  1. What is an operand? List any operands in the above stmt
  2. What is a variable? List any variables in the above stmt
  3. When division is performed in the above stmt, only the operands are considered. Why?
  4. An integer / integer is an integer. Any decimal portion is truncated. What does this mean?
  5. What type should average be declared as? Why?

Mathematical Expressions

- Complex mathematical expressions are created by using multiple operators and grouping symbols
  - expression: programming statement that has value
  - `sum = 21 + 3;`

    expression
  - `number = 3;`

In these two examples, we assign the value of an *expression* to a variable

Arithmetic Operators

- Operators allow us to manipulate data
  - Unary: `operator operand`
  - Binary: `left operand operator right operand`

Operator	Meaning	Type	Example
-	Negation	Unary	`- 5`
=	Assignment	Binary	`rate = 0.05`
*	Multiplication	Binary	`cost * rate`
/	Division	Binary	`cost / 2`
%	Modulus	Binary	`cost % 2`
+	Addition	Binary	`cost + tax`
-	Subtraction	Binary	`total - tax`

Operator Precedence

- `result = 4 * 2 - 3;`
  - `result = ?`
- `result = 12 + 6 / 3;`
  - `result = ?`
- Rules on how to evaluate an arithmetic expression
  1. arithmetic expressions are evaluated left to right
  2. do them in order of precedence
  3. grouping symbols ( )

Operator Precedence

Precedence of Arithmetic Operators (Highest to Lowest)
(unary negation) -
* / %
+ -
(assignment) =

- Operator Associativity
  - If two operators have the same precedence, evaluate them from left to right as they appear in the expression

Practice

```
int x = 3;
double y = 2.5;

cout << 5 + 2 * 3;
cout << ( 10 / 2 – y );
cout << 3 + 12 * 2 - 3;
cout << 4 + 17 / 3.0 + 9;
cout << (6 - y) * 9 / x * 4 - 9;
```

If you are unsure, you can always type up and run the code in Visual Studio

Modulus

- Modulus is the remainder after integer division
- **`remainder = 100 % 20;`**
  - **`remainder = ?`**
- **`remainder = 100 % 30;`**
  - **`remainder = ?`**
- **`remainder = x % n;`**
  - **`remainder = ? (what are the possible values)`**

Problem

- Write a C++ program that allows the user the ability to enter the number of nickels and pennies they have. You are then to print the number of dollars and change that corresponds to.