1.5-1.7: Data Representation and Expressions

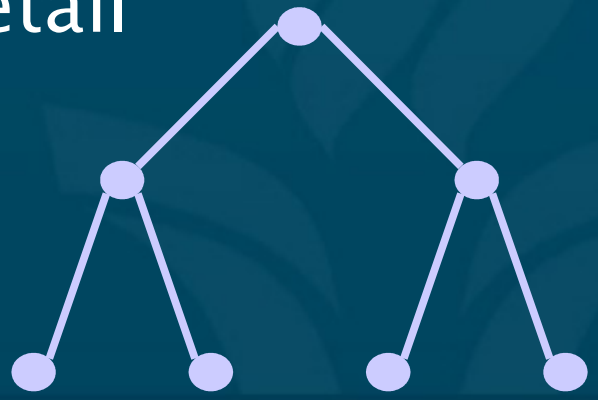
9 Sep 2005
CMPT14x
Dr. Sean Ho
Trinity Western University

Announcements

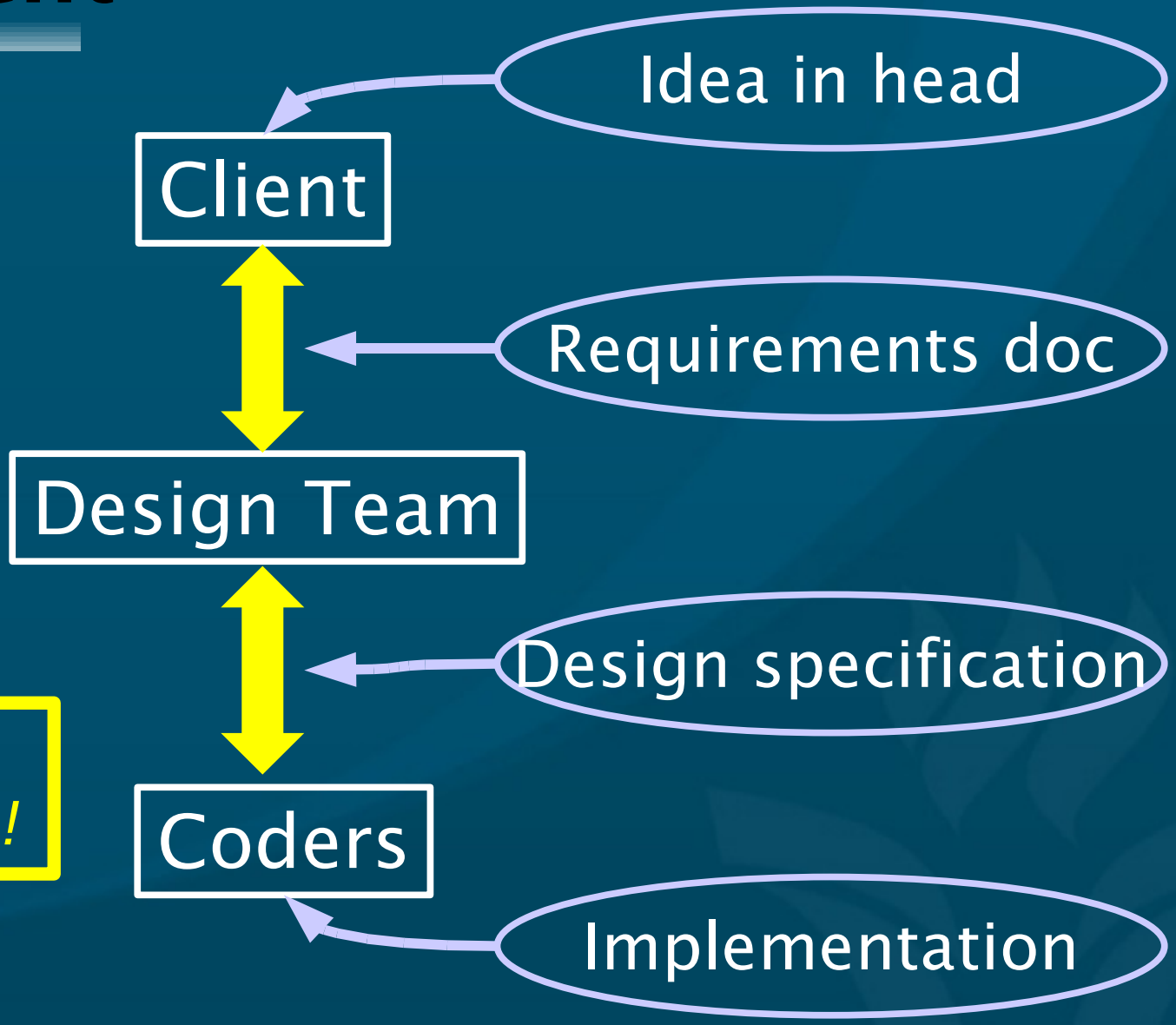
- 141 students: make sure you're signed up for a lab section (MTW 6-10pm)
- Stonybrook software: only licensed for TWU-owned computers (sorry!)
- Short writeups ok for first two labs

Review

- Toolsmiths must know their toolboxes
  - (what does it mean for a computing scientist to be a toolsmith?)
- Top-down vs. bottom-up
- First step in problem-solving? (don't code yet!)
- ***WADES*** *(Write, Apprehend, Design, Execute, Scrutinize)*
- Levels of abstraction / levels of detail



Interfaces in software development



Two-way communication!

What's on for today (1.5-1.7)

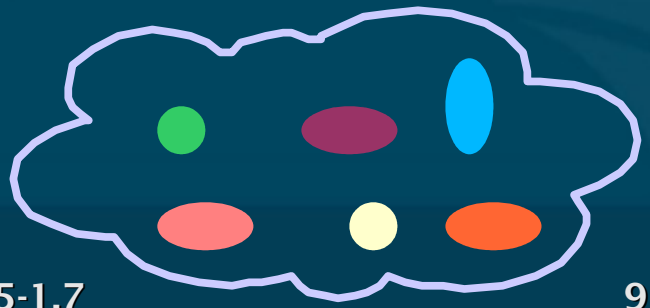
- Data Representation
  - Atomic vs. compound data
  - Data types
  - Abstract Data Types
  - Variables vs. constants
  - Logical operators
  - Operator precedence
  - Expression compatibility

Data representation

- Data vs. information, knowledge vs. wisdom
- Raw data (factoids, memorized mantras) are useless unless you know what they mean!
- "There are 10 kinds of people in the world: those who know binary, and those who don't."
  - (what does "10" mean?)

Atomic vs. compound data

- Atomic: represents a single entity
  - e.g., 8, π, 6.022×10^{23}, z
- Compound: entity that also is a collection of components: e.g.,
  - Set: {43, 5, -29.3}
  - Ordered tuple: (3,9) *(what's the diff from set?)*
  - Complex number: 4.63+2i *(set or tuple?)*
  - Aggregate: (name, age, address, phone#)
- Singleton: {43}



Data types

- Certainly atomic vs. compound data are different types
- But even among atomic data there are types: e.g.
  - Cardinals (unsigned whole numbers; naturals): 0, 1, 2, 3, 4, 5, ...
  - Integers (signed whole): -27, 0, 5, 247
  - Reals: 5.0, -23.0, $3x10^8$
  - Booleans: TRUE, FALSE
  - Characters: 'a', 'H', '5', '='
  - Strings: “Hello World!”, “5”

Different operations for different types (some examples)

- Operators work on operands:
  - e.g. 3+4: operator is "+", operands are 3, 4
- Cardinal type: e.g., +, -, *, /, *print*, etc.
- Character type: e.g., *capitalize*, *print*, etc.
  - 'b' / '4' doesn't make sense
- String type: e.g., *reverse*, *print*, etc.
  - *reverse*(1.3) doesn't make sense
- Array-of-strings type: e.g.,
  - Reverse each string in the array
  - Reverse the order of the array *(different?)*

Go therefore and
make disciples of
all nations, bapti
zing them in the
name of the Fath
er and the Son a

Abstract Data Types

- We define an Abstract Data Type (ADT) as a set of items w/ common properties and operations
  - e.g., Real ADT: reals w/ +, -, *, /, etc.
- Implementation of an ADT:
  - Real-world implementations of ADTs on actual computers have limitations
  - e.g. Can't represent integers bigger than 2147483647 (on a 32-bit machine)
  - e.g. Real (floating-point) numbers can be represented only up to a certain number of significant figures: 1.99999999999 ≠ 2

Variables and constants

- A constant's value remains fixed: e.g., π, e, 2
- A variable's value may change: e.g., x, NumberOfApples
- We can assign new values to variables
  - NumberOfApples = 12
  - NumberOfApples = NumberOfApples – 1
- But not to constants
  - π = 3.0 (can't do this!)
- In Modula-2, we can specify that a particular name is a constant and may not change its value:
  - CONST GodsLove = Infinity;

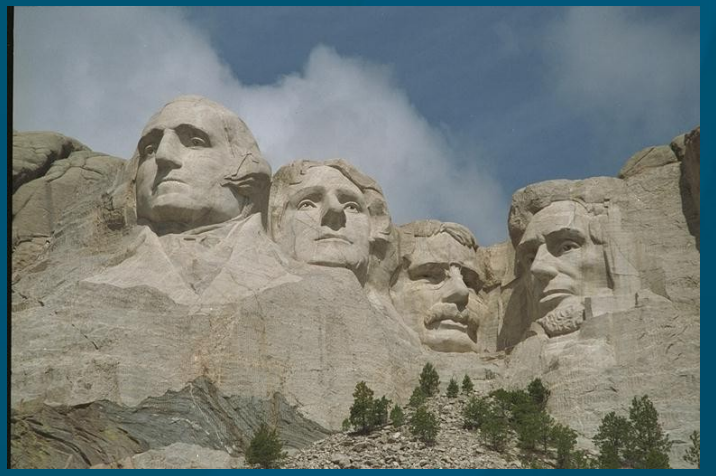
Expressions

- A combination of data items with appropriate operators is called an expression
- Expressions are evaluated to obtain a single numeric result
  - 15 + 9 + 11 + 2 ------evaluation-----> 37
- Operators may evaluate to a different type than their operands:
  - 22.1 > 15.0: what is the type of the operands? What is the type of the result?

Logical operators

- Logical operators are operators on the Boolean type:
  - GodLovesMe = TRUE;
  - ILoveGod = FALSE;
- NOT: flips TRUE to FALSE and vice-versa
  - NOT GodLovesMe ----> FALSE
- AND: evaluates to TRUE if both operands are TRUE
  - GodLovesMe AND ILoveGod ------> FALSE
- OR: evaluates to TRUE if at least one operand is TRUE
  - GodLovesMe OR ILoveGod ------> TRUE

Operator Precedence

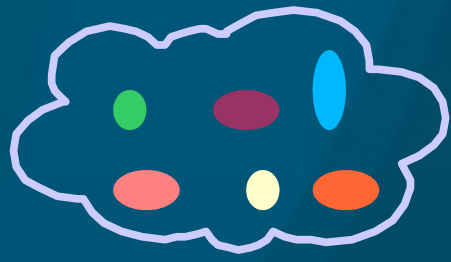


- How would you evaluate this?
  - 5 + 4 * 2
  - (5 + 4) * 2 = 18: Addition first
  - 5 + (4 * 2) = 13: Multiplication first
- Precedence is a convention for which operators get evaluated first (higher precedence)
  - Usually multiplication has higher precedence than addition
- When in doubt, use parentheses!

Expression compatibility

- 5 + TRUE doesn't make sense: incompatible types
- What about 5(cardinal) + 2.3(real)?
  - Works even though they're different types, because the two types are expression compatible
- The "+" operator is overloaded:
  - It works for multiple types: both cardinals and reals
- 5(cardinal) + (-2)(integer) works, too

Review of today (1.5-1.7)



- Atomic vs. compound data (examples?)
- Data types (examples?)
  - What's the difference: 5, 5.0, '5', “5”, (5), {5}
- Operators, operands, ADTs, implementations
- Variables vs. constants
- Logical operators: NOT, AND, OR
- Operator precedence
- Expression compatibility (what types?)

=	NOT	*
AND	/	<
+	OR	-

TODO items

- Sign-up for a lab section (MTW 6-10pm)
- Buy 14x coursepack (vols1-2) from bookstore
  - Or borrow from a previous CMPT14x student
  - Read through 2.1 for Monday
- Go to Neu9 computer lab:
  - Make sure you can login
  - Stonybrook intro on course www (due Wed)
- Ch1 quiz next Monday start of class
- Ch1 homework due next Wed