

# CS603 Programming Language Organization

---

Smalltalk

Spring 2005

Department of Computer Science

# Overview

- Questions
- Smalltalk the Language ( $\mu$ -Smalltalk)
- Examples
- Reading for next time

# Smalltalk (Real)

Smalltalk Term	C++ term	Smalltalk meaning
Class	Class	Encapsulates the representation of a type of data
Object	Object	
Method	member function	An operation defined in a class
Message	member function name	
Message send	Function call	
Instance Variables	data member	The names of the values representing objects of a class
Protocol		Messages to which an object responds

# Smalltalk (Real)

- Further differences from C++
  - Everything is an object
    - Integers
    - loops and other control constructs!
  - All methods belong to a class
  - There is a single inheritance hierarchy
    - rooted at Object

# Smalltalk (Real)

- Further differences from C++ (cont.)
  - Classes are objects
  - Every object is an instance of a class
  - `self == this`
  - Objects created by sending messages to class

# Smalltalk ( $\mu$ -Smalltalk)

toplevel	expression   use   val   fundef   classdef
classdef	(class class class inst-vars methoddef+ )
inst-vars	( variable* )
methoddef	( {method   classmethod } name (variable* ) expression )
fundef	same as Chapter 2
expression	literal   variable-name   (set variable-name exp)   (begin {exp})   (message-name exp {exp})   (block ( {argument-name} ) {exp})   [ {exp} ]
literal	integer   symbol   # ( {array-element} )
symbol	#name
class	name

# Example Smalltalk ( $\mu$ -Smalltalk)

- Financial History – a classic example from “Blue Book” (Figure 7.1)

```
(class FinancialHistory Object
  (cashOnHand incomes expenditures)
  (method setInitialBalance: (amount)
    (begin
      (set cashOnHand amount)
      (set incomes      (new Dictionary))
      (set expenditures (new Dictionary)))
    self))
)
```

# Example Smalltalk ( $\mu$ -Smalltalk)

- But how do we create an object of FinancialHistory?
- Real Smalltalk: send message to FinancialHistory class
- $\mu$ -Smalltalk: the same – use class method to create an instance of FinancialHistory

— `(val account (initialBalance: FinancialHistory 1000))`