Initialization and Inheritance

Smalltalk.Core defineClass: #Parent
  superclass: #{Core.Object}
  indexedType: #none
  private: false
  instanceVariableNames: 'foo '
  classInstanceVariableNames: '
  imports: ''
  category: 'CS535'

Class Method

new
  ^super new initialize

Instance Methods

initialize
  foo := 6.

foo
  ^foo
Initialization of Subclass

How to initialize bar?

Smalltalk.Core defineClass: #Child
   superclass: #{Core.Parent}
   indexedType: #none
   private: false
   instanceVariableNames: 'bar '
   classInstanceVariableNames: '
   imports: '
   category: 'CS535'

   **Bad Idea 1 – Use Same pattern**

Child class>>new
   ^super new initialize

Child>>initialize
   bar := 2.

Child>>bar
   ^bar
Why bad?

- Does not work!

| test |
test := Child new.
test foo "returns nil"

- initialize is called twice

Child class>>new is not needed
Child class inherits an identical method
Bad Idea 2 – Subclass initializes Parent Variable

Child>>initialize
   bar := 2.
   foo := 6.

Child>>bar
   ^bar

Why Bad?

Child class now involved in private affairs of the Parent

Changes to the Parent instance variables require changing Child
Solution

Child>>initialize
  super initialize
  bar := 2.

Child>>bar
  ^bar