

CPSC 256

Smalltalk

Teams: None, please work individually on this project

References:

1. Pointers Memory lectures and projects
2. Classes, Objects lectures and projects
3. ‘Vector full of Base Class Pointers’ example project

Sample Code:

See 256_proj_smalltalk

Topics covered by this project;

- Using pointers to manipulate objects
- Using vectors to hold objects and pointers
- Class Hierarchies
- Abstract Base Classes
- Polymorphism

Class Hierarchy

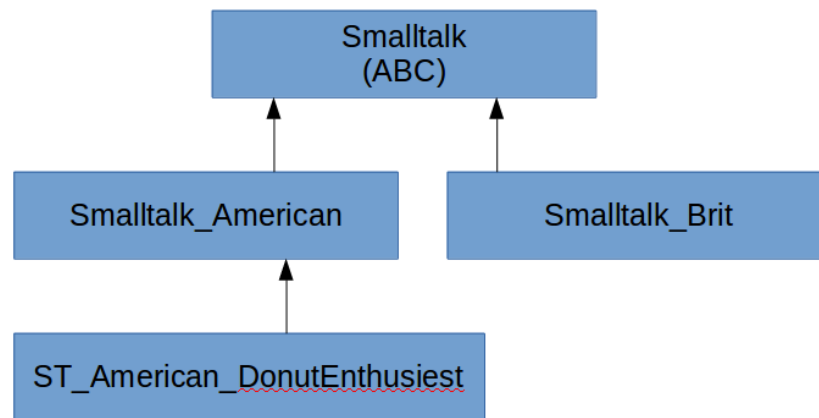
You are developing a class hierarchy for this project. An Abstract Base Class (ABC), ‘Smalltalk’ defines the hierarchy behavior. Most of your work in this class structure will take place in Smalltalk.

Classes derived from Smalltalk must implement `populatePhrases()`. A function that initializes the baseclass vector with phrases that are unique to that class type. For instance, `Smalltalk_American` will populate its internal vector of strings with the american phrases found in `constants.h`.

Please also provide implementations for functions declared in `Functions.h`

Inheritance Diagram

All classes inherit publicly. The class hierarchy is as follows;



I have given you the header files and some of the implementation.

Testing

Please see 256_proj_ST.cpp. This file contains a complete tester for the classes and functions you develop.

Submission:

You will do your work in the following files. Please submit just these files. Please do not zip them together, or embed them in a directory structure. Just the 5 files.

Functions.cpp

Smalltalk_American.cpp

Smalltalk_Brit.cpp

Smalltalk.cpp

ST_American_DonutEnthusiest.cpp

Grading:

For each concrete class remember to push as much common functionality as possible into base classes! This cuts down on repetitive code in derived classes.

Your grade is as noted in the tester with the following possible deductions:

5% Submission instructions not followed

10% valgrind returns errors

10% Code style problems(refactor repetitive code into functions, magic numbers, large blocks of empty space, large chunks of commented out code, not pushing as much functionality in to the base class as possible, lack of appropriate comments, etc.)