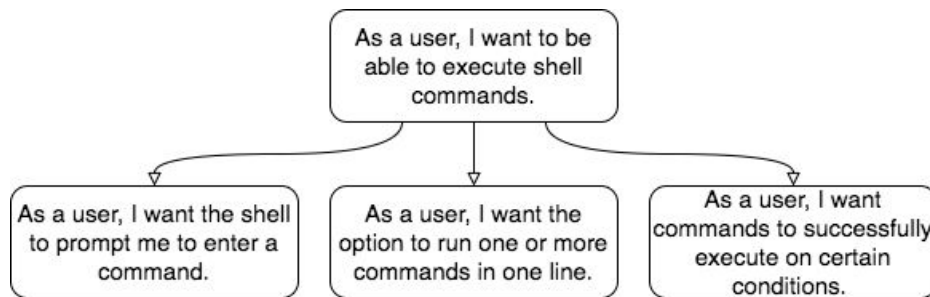# Assignment 1: Design

Eric Ong (861267345)
Garret Lim (861267836)
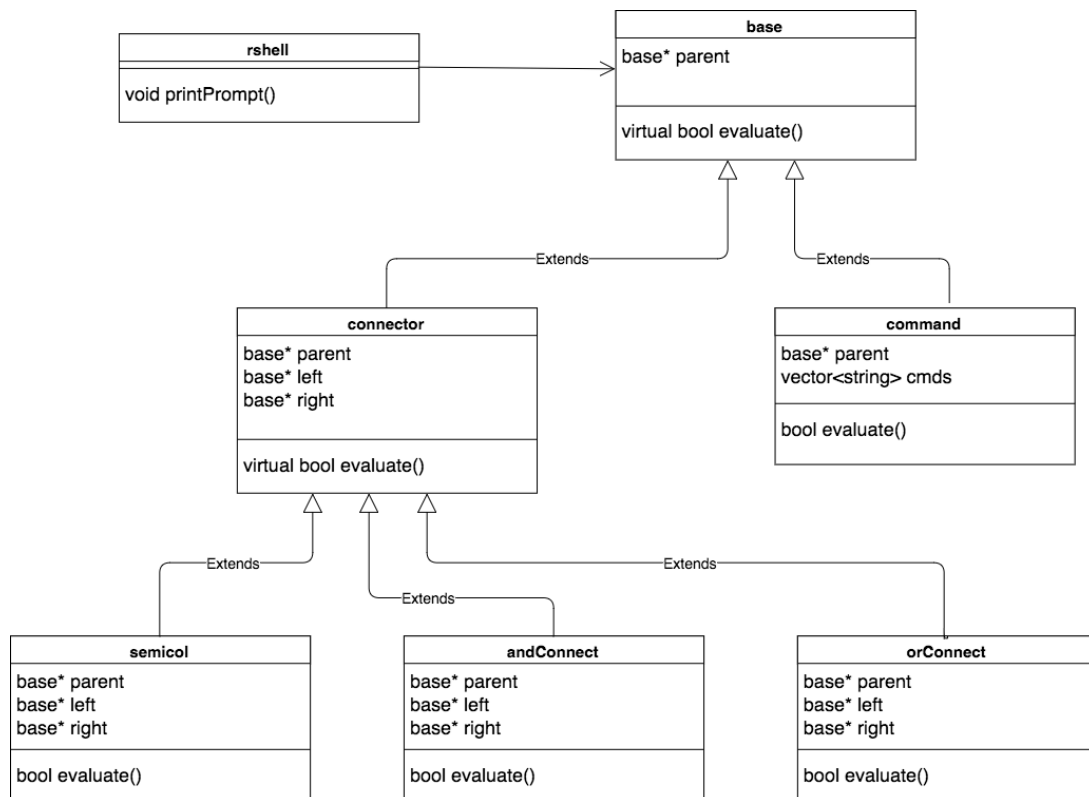
Winter 2018
Due: 1/26/18

Introduction

We are building a basic command shell that reads in a command in a command prompt and executes them. The user has an option to execute more than one command at a time per line by using connectors ('&&', '||', ';') or by separating each command by a semicolon (;). Below is an epic illustrating the features desired for the program.

UML Diagram

Classes/Class Groups

rshell

rshell is what prompts the user to enter a command (by displaying a '$') in one line. The commands the user enters are read in and then executed.

base

The **base** class acts as a foundation for the other classes involved in the program. The base contains a virtual function, which the other classes like the connectors and commands will inherit, to evaluate/execute the command passed in. The base pointer "parent" is what connects the expressions together, since we plan on implementing the program with a tree.

connector

The **connector** class encompasses the class group of 3 different connectors used in the program ("&&", "||", and ";"). It contains a left and right base pointer in order to form a larger expression by connecting the expressions that are to the left and right of it.

command

The command class is used to store user inputs that are meant to execute a program. The inputs are stored in a vector of strings.

semicol

The **semicol** class is one of the classes in the class group that inherits from connector. It is represented with the symbol ";". In the evaluate function of this class, all command following this object will be executed.

andConnect

The **andConnect** class is also one of the classes in the class group that inherits from connector. It is represented with the symbol "&&". In the evaluate function of this class, the command following this object will be executed if the command preceding this object succeeded.

orConnect

The **orConnect** class is another class in the class group that inherits from connector. It is represented with the symbol "||". In the evaluate function of this class, the command following this object will be executed if the command preceding this object failed.

Coding Strategy

We will collaborate using Github and work on our separate branches for our individual tasks. We will then merge branches to test if the code functions correctly. We will use Kanban to manage our workflow as well as hold physical meetings to make sure both members are keeping up.

Roadblocks

Some roadblocks we may encounter include:
- Difficulty understanding the assignment. Although we may organize a plan and workflow in preparation for the assignment, some concepts and specifications in the assignment may be unclear. This prevents us from proceeding with our work , which would possibly cause us to lag behind our schedule.
- Either partner not being on the same "page", or a lack in communication between the two. In other words, one partner may work ahead of the other and vice versa, therefore resulting in an inconsistency or lack of awareness in workflow.
- Technical issues may arise along the way. For instance, GitHub's servers may be down, preventing us from pushing our code, or one of our laptops/machines may crash.