Matthew Zachary

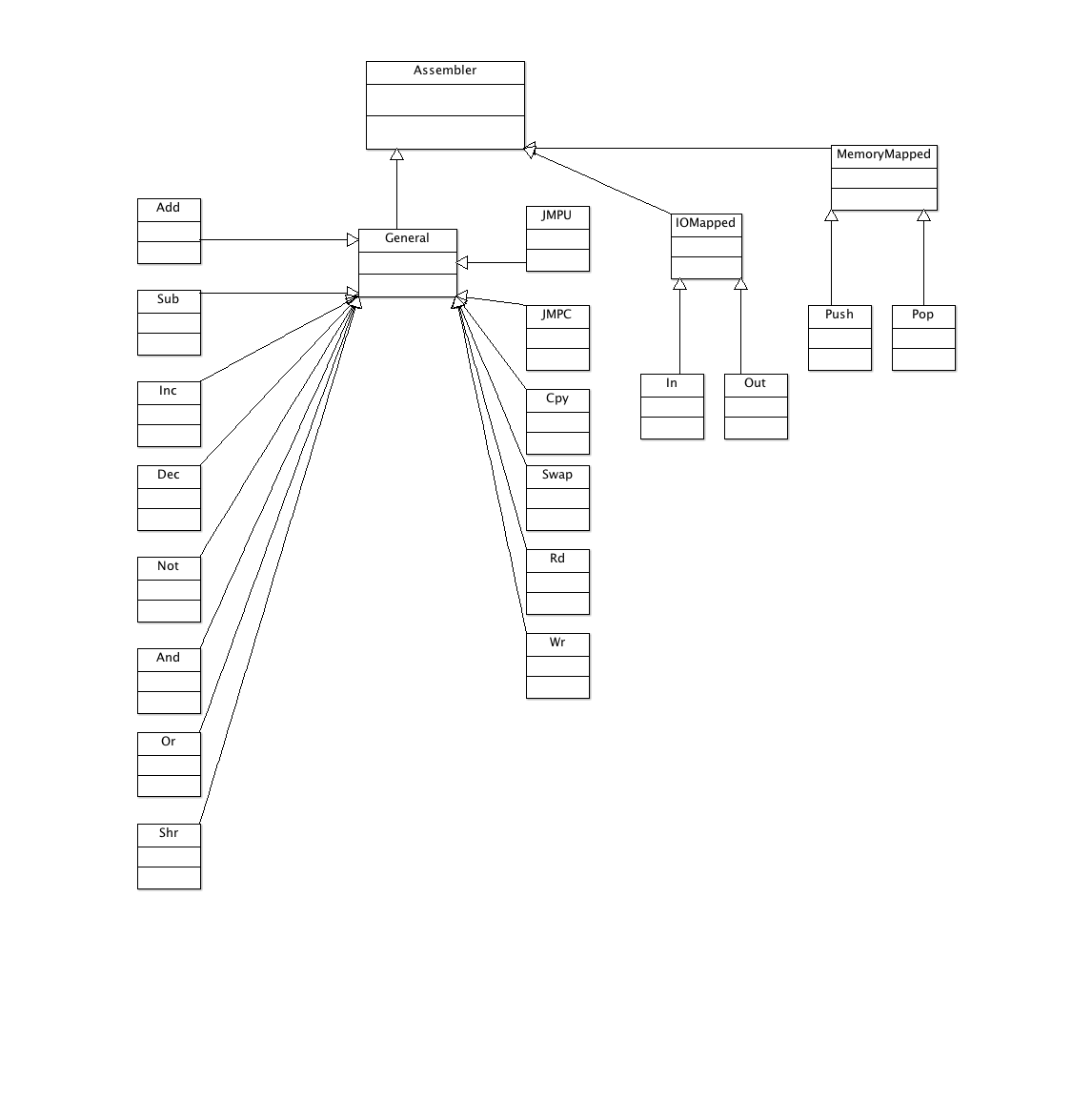
Connor Goldberg

Final Project Proposal

Overview:

For this project we will be implementing an assembler that will take an input file of assembly code and create a new file with machine code. We will create the assembly syntax just for the purpose of this project. The assembler will be able to produce machine code for more than one architecture depending on the assembly code. Within the assembly code there will be space for declaring which type of architecture the user will be using. The program itself will take this into account as it builds the machine code. To run the program the user will execute the program in the command line. The program will ask the user for name of the file, and in turn will output any errors the assembler has, or will confirm the success of the program.

UML:



* Classes – Each instruction has it’s own class
* Polymorphism – Each architecture has its own abstract class
* Operator Overload – We will at least overload the instream to assign memory address data to a class.
* Data Structures – We will be using a map to store memory address and variable data.
* Exception Handling – Used to output errors found in the assembly code input.
* File Processing – Used to input and output the assembly code / machine code, respectively.