Project Solution Explanation

So, in order to implement the MIPs instruction set, we first need to understand how to run each individual line of code consecutively. We would need to parse each line into an array so that we can process each operation and operand of each instruction, then actually computing the function.

Then, for branching, we need to keep track of labels. So, when parsing the data, we look for the ":" character and take the string before it, then store it into a labels dictionary, separate from the instructions array. The data structure for labels need to be a dictionary because when processing branching operations such as beq $t1 $t2 loop, "loop" needs to be found somewhere and there needs to be an associated value to that label name. Using an array would only give us the label's name but not its value, and dynamically creating global variables would be a waste of resources. We use the associated value to change the current instruction to the instruction where the label is, which either moves the current instruction forward or backwards and continues the code. Therefore, we must utilize a dictionary rather than an array.

I used an object oriented approach to create the MIPS simulator so that I could easily set the register values and execute internal instructions as needed. This way the code is more readable and possibly able to be used in conjunction with other programs as a separate library