- Authors:
  - Cameron Dunn, dunnc11@mcmaster.ca
  - Rafeed Iqbal, iqbalr8@mcmaster.ca
- Group ID on Avenue: 23
- Gitlab URL: https://gitlab.cas.mcmaster.ca/dunnc11/l-3-assembly-group-23

Self Reflections

Cameron Dunn

• How much did you know about the subject before we started? (backward)

A lot of what we are doing this lab is similar to what we've done before (Assembly coding and instruction architecture in 2GA3, software design and object-oriented design patterns from 2AA4). Going into the lab I somewhat had an idea on how compilers worked, but after having completed the lab I have to say there is a lot I did not know. The lab served as a good guideline for how the Visitor pattern worked, and how high level code can be translated into assembly code.

• What did/do you find frustrating about this assignment? (inward)

I found F4 and F5 to be the most frustrating. F2 and F3 could be mostly done by making slight changes to the code that was given to us, but F4 and F5 took both a lot of research beforehand, and a lot of trial and error during the coding process. There were a lot of aspects and edge cases which had to be considered, and I had to examine both the Pep9 documentation and the ASTs generated very carefully before being able to code up an aplicable solution.

• If you were the instructor, what comments would you make about this piece? (outward)

If I were the instructor, I would have to say that it was expected that the work be done incrementally, and thus better time management is required. It would also add that the code should be optimal, and the visitor and generator patterns should be better utilized.

• What would you change if you had a chance to do this assignment over again? (forward)

I would attempt to better manage how the work was divided. While we did work on the project incrementally, much of the progress was done towards the end, which may have affected the quality of the work being submitted. Additionally, I would have liked to have attempted the bonus problems, as I found being able to actually implement features was very fulfilling after overcoming the frustration.

Rafeed Iqbal

• How much did you know about the subject before we started? (backward)

I had a basic idea of computer architecture from 2GA3 from last year, as well as an idea of how to translate basic programs into ARM assembly code. However, most of this lab was entirely new to me. We had learned about Object oriented design patterns in 2AA4 but had not learned the visitor pattern. The example code provided helped me gain an understanding of how visitors and generators worked, as well as how python code is structured through the AST. Overall I did not know much at the beginning of the lab, but I have learnt a lot over the course of it.

• What did/do you find frustrating about this assignment? (inward)

I found the Pep9 documentation to be lacking, and found myself having to scour a myriad of documents and pdfs to find an explanation for the problems I was facing. This was especially apparent when working on F5, as it was extremely confusing for me on how to properly allocate memory to the arrays, and how to differentiate between local and global declarations of arrays and how to handle the memory de-allocation.

• If you were the instructor, what comments would you make about this piece? (outward)

We were given a large period of time to complete the lab, and it was meant to be done incrementally. This somewhat reflects real-world projects, where features are added over time, and work is divided reasonably. If I were the instructor I would say that it is very important to manage your time, and work on reasonable portions of work at a time. One should take a break to reflect on what progress has been made as well as take time to test thoroughly what has been implemented. This can help catch bugs, and reconcile what was asked of us with what we delivered.

• What would you change if you had a chance to do this assignment over again? (forward)

One thing I would change is to make the generated code readable, the current hashing function makes the variable names unreadable, but is necessary to facilitate the translation of names larger than the 8-character limit imposed by Pep9. If possible I would like to add some way to add comments beside the code, to make it clear what is happening in each section of the generated code. Another thing I would have liked to do is implement all the features mentioned in the bonus part of the lab, but I was not even able to attempt it due to lack of time.