

Compiler Project PA5 – Documentation

Due: 7/25/24 11:59:59pm

The final submission of the project adds no new functionality, but provides an opportunity to correct PA1-PA4 errors in your compiler and make improvements as you wish. It also provides an opportunity to add extensions to your compiler for extra credit. The extra credit is not needed – all grades can be obtained with just the basic project.

You need to submit a guide for your compiler, preferably as a PDF. Turn in all project files on the “PA5—Final Compiler Submission” assignment on Gradescope.

REQUIRED ITEMS: GUIDE, ASTChanges.txt, SOURCE CODE

Recall how x64 documentation was difficult to read. Try not to repeat those mistakes in your guide. You do not have to restate the obvious. Assume your reader is generally familiar with compilers, doesn't need to know what is an AST, and doesn't need to know the specifics of bytecode generation. Don't restate what you already did in ASTChanges.txt.

YOUR GUIDE IS NOT A JUSTIFICATION FOR DESIGN DECISIONS!

Instead, it is a guide. In as few sentences as possible, concisely describe how you handled the parts of the compiler.

- 1) Syntactic Analysis- Did you use recursive descent or a PDA? Did you make everything a token, or minimize the number of tokens? Anything else?
- 2) AST Generation- “We assume general familiarity with ASTs. See ASTChanges.txt for data beyond syntax that is stored in ASTs.”
- 3) Contextual Analysis- Did you do one traversal or two? Where is SI done?
- 4) Code Generation- Did you do optimization? How is your memory laid out?
- 5) Finally: Any greedy decisions to watch out for? For example, if you made everything 64-bit even though JLS requires int to be 32-bit, then how did you make sure add/multiply/subtract did not cause issues? If there are indeed parts of your compiler where you simply did not handle errors, that is fine, but mention those errors so that someone else can take over your Compiler.

There is no minimum page count, paragraph count, word count, etc. Your guide could even entirely be drawn in MSPaint. The most important part is that it conveys how your compiler is organized. Please do not submit more than a page.

If you have a GitHub, a good idea is to do documentation in the README.md file.