

CS4TB3: Compilers and Syntax-Based Tools

Department of Computing and Software
McMaster University
Winter 2018

February 10, 2018

Instructions

This project is to be completed with your assigned group. Each member of the group is expected to contribute equally to final deliverables. The complete project is worth **20%** of your final grade. The complete project is divided into a set of deliverables as listed below. The course policies as listed at (<https://cas.mcmaster.ca/~cs4tb3>) apply.

1 Executive Summary (Due Thurs. Mar 1, 3:00 pm) 10%

Within your group, select a research topic and write a half page summary describing what your group will be writing about. This should state the scope of your activity and give the reader a clear idea of what will be contained in your final deliverable. The summary should at least contain the following information.

- *a high level description of your topic*
- *an outline of the area(s) within your chosen topic that you will focus on*, After reading this summary, the audience should know what to expect in your project. She should quickly be able to determine what is contained (and possibly what is not) in your project and if it is of interest.
- *a description of implementaton* The project requires an implentation to support your report. You should give a high-level overview of the implementation's functionality and state how it fits in to the project's scope
- *relevance to field*, The summary should also tell the reader how the topic and your chosen focus is related to the computing field at large. A reader at your level (such as one of your peers), should see the topic's relevance.

Here are some possible high level topics. You are not bound to these, but the topic must related to either *Compiler Construction* or *Language Theory*.

1. generating assembly code for hardware platforms (ARM, Intel)
2. generating intermediate code (JVM .class files, CLR, LLVM)
3. implementing specific optimizations for speed, memory, battery life
4. static program analysis
5. code for concurrency, exception handling or other specific language features
6. using functional languages for compiler construction
7. functional language vs procedural language code generation
8. partial compilation, JIT techniques
9. parsing techniques (not covered in class)
10. approaches to garbage collection

2 Report (Due Mon. Apr 9, 3:00 pm) 40%

Write the report based on the promises made in your **Executive Summary**. The first page should be a revision of the Executive Summary from Step 1. It should be approximately 15-20 pages, one and one-half (1.5x) spacing. **It must be delivered in .pdf format.** The work will be graded on the following.

- **introduction.** This should provide a “big picture look” at your topic and present an outline of the rest of the paper.
- **content.** The “meat” of your work. should provide technical detail but be presented in a way such that developers not familiar with the area can understand it. Work should progress logically and seamlessly from one topic to another. *Beware - this can be difficult to accomplish when there are multiple writers.* Ensure your group members get it done early enough to proof read the document as a whole.
- **discussion.** This section is reserved to provide commentary, observations or interesting findings about your work. Also, speak about anything that couldn’t be completed for whatever reason.
- **conclusion.** This should summarize the whole work. Nothing new about project should be stated here but of interest are any potential next steps for others to build on your work.
- **spelling/grammar.** Papers at this level should be virtually error-free.
- **references.** There should be sufficient quantity and quality. Quality references are peer-reviewed journals, published books and web-pages of large established companies or research colleges/universities. Personal blogs and wikis are not seen to be reliable sources. A variety of different sources is ideal. References should be cited according to a recognized standard (either APA or MLA is fine).
- **graphical content,** High quality pictures or graphs to help illustrate the paper’s ideas should be included where appropriate.

3 Prototype (Due Thurs. Mar 22, 3:00pm) 30%

Your project will include an implementation which helps illustrate the ideas expressed in your paper. Source code and accompanying files are to be submitted on Avenue in compressed zip format. Do not submit executable or intermediate files. The reviewer should be able to build your program using instructions included. If you use Github or another repository system to store your files, you can submit a project link in Avenue, in lieu of the actual project files, as long as it is publically accessible.

The implementation must be written in a language of your choice, but must follow typical software engineering practices. It should be well-commented, decomposed into modules (where appropriate) and easily testable.

- **functionality.** Does the choice of implementation complement the purpose of the report?
- **correctness.** Does it provide the functionality promised in the proposal?
- **design.** Is the design well-thoughtout? Is the work well documented?
- **testability** Can another student *build, run and test* your program with your documentation? Have you provided sample data so the program can run without additional work?

4 Presentation (Friday Mar 23, Apr 6) 20%

Your group will present a summary of the resulting research to your peers. It does not have to follow the exact structure of the paper but it should present the same main ideas. A demonstration of your prototype should be included. **It should not exceed 20 minutes.** You can assume the audience is at the same level you are (i.e. senior year computer science students). Work here will be graded on the following.

- **visual aids.** Slides, screen output or anything else you find useful
- **style, approach.** Can I hear the speaker? Is the presentation easy to follow?
- **content** Did the speakers present the important parts of the topic?