EN.605. 615 section 8VL

Compiler Design with LLVM

Course Information

Compiler Design with LLVM

EN.605. 615 8VL (3.0 Credits)
Spring 2022 [AE Spring 2022]

Description

The components of a compiler appear in every software application that handles input from an external source. This course shows how the components of a compiler are built and how they fit together to extract meaning from the input and how the data flows through the compiler's components to become useful to applications. Students will get practical experience in how to use the LLVM tools to build a complete compiler for a subset of the C++ programming language that can target almost any platform. Students will also get experience in developing a "Just In Time" component for an application that will accept code as input into the application while it is running, to be compiled and linked into the application so the application can execute it. Prerequisites: This course has no formal prerequisites, but experience with C++ is highly recommended because LLVM is written in C++, and therefore, all homework will be in C++, and this course is software homework intensive.

Department: PE Computer Science
College: Engineering and Applied Science Programs for Professionals

Instructor

Allyn Shell
ashell2@jhu.edu

Teaching Assistant :

None

Class Times :

The class will meet virtually once a week at 7:15 PM EST on Thursday evenings starting January 27, 2022.

Course Location :

This course was a face-to-face course which (due to Covid-19) has transitioned to meet remotely using Zoom.

Communication Policy :

I prefer that students contact me via **email**. Please be sure to include the course number in the subject line. I will make every effort to respond to your inquiry within 24 hours (except Sunday).

Office Hours :

Since this course meets virtually every week, there will be no separate Office Hour. If you have any question on anything you may contact me via email. I will make an effort to be responsive to email inquiries within a day. Those inquiries that are personal in nature will be handled discretely. If an email identifies an error in the course material, the correction will be published to the whole class as quickly as possible. You may also raise questions at any time during the virtual sessions.

Course Structure :

The course material will be presented by the instructor virtually via Zoom. The classes will be recorded for review by students, but class attendance is required since quizzes and homework assignments will be handed out via email while meeting virtually. Class participation will contribute to the grade.

Course Topics :

- LLVM Environment
- Introduction & Overview
  - Fundamental Definitions
  - Compiler Organization
  - Chomsky Hierarchy
  - Intro to LLVM
  - ASCII
- Scanning
  - Intro to Lexical Analysis
  - Finite State Machine
  - Regular Expressions
  - Example Scanner
- Precompiler
- Parsing
  - Parse Tree and Abstract Syntax Tree (AST)
  - Backus-Naur Form (BNF) and Extended BNF (EBNF)
  - LL and LR Parsing
  - Left Recursion
  - Recursive Decent Parsing
  - AST generation
  - First and Follow sets
  - LL(1) Parsing
  - Error Recovery
  - LR Parsing intro
  - LR Parsing Errors
- Semantic Analysis
  - Symbol Tables
- LLVM IR generation
  - Intro
  - IR organization, Simple
  - IR for Function calls
  - SSA & Phi Functions
- Optimizations.
  - Intro
  - Optimize Passes
  - Analysis passes
  - Pass Manager
- Target Code Generation
  - Runtime Environments
  - Stack Activation Records
  - Kinds of Software Languages
- Linkage Editor
  - Link Time Optimization
  - Relocatable Object Module
  - Loader
- Dynamic Link & JIT
- Miscellaneous Topics

- Cross compilation
- P-Code
- Virtual Machines
- Threaded VM Code
- etc.

Course Goals :

This course is designed to provide **practical** experience adapting parts of a compiler to many different types of applications. The student will learn about information structure, programming language translation and compiler design concepts.

Course Learning Outcomes (CLOs) :

- You should be able to understand the structure of information.
- You should be able to understand the essence of computation
- You should be able to understand practical compiler organization and implementation.
- You should be able to understand finite state machines and how to use them for lexical scanning.
- You should be able to understand regular expressions and context free grammars.
- You should be able to understand front end techniques for parsing and production of abstract syntax trees and symbol tables.
- You should be able to understand LLVM’s intermediate code representation.
- You should know how to build a recursive descent parser.
- You should be able to understand the organization of LLVM.
- You should be able to understand how to use LLVM Opt subsystem to produce optimized code.

- You should be able to understand how to build your own compiler.

- You should be able to understand how to adapt compiler components for use within many other applications.

Required Text and Other Materials

Textbooks :

Compiler Construction: Principles and Practice, **Kenneth C. Louden**, PWS Publishing Company, 1997, ISBN 0-534-93972-4.

Other Materials & Online Resources :

The Definitive ANTLR 4 Reference, by **Terence Parr**, The Pragmatic Bookshelf, 2013. **ISBN:** 978-1934356999

Required Software :

LLVM Pre-Build Binaries for **Release 13.0.0** for your preferred platform.

LLVM/llvm-project on GitHub, ***https://github.com/llvm/llvm-project.*** **(**You may need to build the C++ libraries for your preferred platform. See **Documentation**.**)**

Kaleidoscope: Implementing a Language with LLVM, part of the LLVM Compiler Infrastructure ***llvm.org***, **Release 13.0.0** Documentation, available online at ***http://releases.llvm.org/13.0.0/docs/tutorial/index.html.***

ANTLR4 by Terrance Parr, available from ***https://github.com/antlr/antlr4***. The online documentation for ANTLR4 is available at ***https://github.com/antlr/antlr4/blob/master/doc/index.md***.

Technical Requirements :

You should refer to **Support** on the course menu for a general listing of all the course technical requirements.

Evaluation and Grading

Student Coursework Requirements :

1. Quizzes and Classroom Participation (10% of Grade)

   This includes quizzes, classroom Q&A participation and classroom activities.

2. Homework Software Projects (65% of Grade)

This class has two software projects: Completed project code must be submitted in Blackboard.

- a **Spreadsheet Project** (skeleton provided) that uses your hand made front end compiler components and **LLVM Just In Time** components to accept code input and to compile, link and execute the input code (35%), and
- a **Compiler Project** built with LLVM components (30%).

And, a research project to be presented as an informal paper and as an in class presentation.

- an Optimization **Research Project** (counted as part of the Compiler Project)

3. Exams (25% of Grade)

A mid-term (15%) and a final (10%) will be given. They will be done in class. Students will have three hours to complete each exam.

Grading Policy :

Late homework will be penalized promptly at 7:15 PM. Work that is partially complete will be given credit for the part completed and submitted on time. Only the incomplete part will be penalized based on when it is turned in. Grades will be posted in Blackboard within one week after assignment due dates.

EP uses a +/- grading system (see "Grading System", *Graduate Programs* catalog, p. 10).

Score Range	Letter Grade
100-98 %	= A+
97-94 %	= A
93-90 %	= A−
89-87 %	= B+
86-83 %	= B
82-80 %	= B−
79-77 %	= C+
76-73 %	= C
72-70 %	= C−
69-67 %	= D+
66-63 %	= D
<63 %	= F

Course Evaluation

Course Evaluation :

Students will be given opportunity to evaluate the course and the instructor's presentation after the midterm exam and after the final exam.

Policies

Course Policies :

This course is homework intensive. Do not get behind. It is very difficult to catch up.

It is recognized that many students have full time jobs that have deadlines and sometimes travel. If you know you are going to have a conflict with your work schedule, **you may request accommodations before the fact**. If you wait until **after the fact**, the answer is already **NO**. If you have a medical emergency and want to request consideration after the fact, your physician will need to provide documentation to validate the events.

If you find that you cannot keep up due to work pressures or medical issues, you may request an **incomplete** grade. This is **NOT** granted automatically. It will require some acceptable, documented justification. An incomplete grade will allow you to spread out the work on a schedule agreed to by you, the instructor and the department chair.

If you cannot complete the homework you can also **audit** the course without credit. This will allow you to acquire all of the course materials that you have paid for, but it will not count toward (or against) a degree.

Additional Resources :

Personal Wellbeing
If you are struggling with anxiety, stress, depression or other mental health related concerns, please consider connecting with the Johns Hopkins Student Assistance Program (JHSAP). If you are concerned about a friend, please encourage that person to seek out our services. JHSAP can be reached at 443-287-7000 or https://jhsap.org/

Tutoring Website

Johns Hopkins Engineering for Professionals offers a tutoring connection network that allows students to connect with other Johns Hopkins Engineering students or alumni for tutoring services. This service allows students to search a list of courses to “Find a Tutor” or complete a profile to “Become a Tutor.” More information about this service can be found on the tutoring website (https://tutor.ep.jhu.edu/).

Deadlines for Adding, Dropping and Withdrawing from Courses

Students may add a course up to one week after the start of the term for that particular course. Students may drop courses according to the drop deadlines outlined in the EP academic calendar (https://ep.jhu.edu/student-services/academic-calendar/). Between the 6th week of the class and prior to the final withdrawal deadline, a student may withdraw from a course with a W on their academic record. A record of the course will remain on the academic record with a W appearing in the grade column to indicate that the student registered and withdrew from the course.

Academic Misconduct Policy

All students are required to read, know, and comply with the Johns Hopkins University Krieger School of Arts and Sciences (KSAS) / Whiting School of Engineering (WSE) Procedures for Handling Allegations of Misconduct by Full-Time and Part-Time Graduate Students.

This policy prohibits academic misconduct, including but not limited to the following: cheating or facilitating cheating; plagiarism; reuse of assignments; unauthorized collaboration; alteration of graded assignments; and unfair competition. Course materials (old assignments, texts, or examinations, etc.) should not be shared unless authorized by the course instructor. Any questions related to this policy should be directed to EP's academic integrity officer at ep-academic-integrity@jhu.edu.

Students with Disabilities - Accommodations and Accessibility

Johns Hopkins University values diversity and inclusion. We are committed to providing welcoming, equitable, and accessible educational experiences for all students. Students with disabilities (including those with psychological conditions, medical conditions and temporary disabilities) can request accommodations for this course by providing an Accommodation Letter issued by Student Disability Services (SDS). Please request accommodations for this course as early as possible to provide time for effective communication and arrangements.

For further information or to start the process of requesting accommodations, please contact Student Disability Services at Engineering for Professionals, ep-disability-svcs@jhu.edu.

Student Conduct Code

The fundamental purpose of the JHU regulation of student conduct is to promote and to protect the health, safety, welfare, property, and rights of all members of the University community as well as to promote the orderly operation of the University and to safeguard its property and facilities. As members of the University community, students accept certain responsibilities which support the educational mission and create an environment in which all students are afforded the same opportunity to succeed academically.

For a full description of the code please visit the following website: https://studentaffairs.jhu.edu/policies-guidelines/student-code/

Classroom Climate

JHU is committed to creating a classroom environment that values the diversity of experiences and perspectives that all students bring. Everyone has the right to be treated with dignity and respect. Fostering an inclusive climate is important. Research and experience show that students who interact with peers who are different from themselves learn new things and experience tangible educational outcomes. At no time in this learning process should someone be singled out or treated unequally on the basis of any seen or unseen part of their identity.

If you have concerns in this course about harassment, discrimination, or any unequal treatment, or if you seek accommodations or resources, please reach out to the course instructor directly. Reporting will never impact your course grade. You may also share concerns with your program chair, the Assistant Dean for Diversity and Inclusion, or the Office of Institutional Equity. In handling reports, people will protect your privacy as much as possible, but faculty and staff are required to officially report information for some cases (e.g. sexual harassment).

Course Schedule

Course Schedule :

Module	Date	Module Title	Assignments
	see Course Outline		