



MICHAEL AND CECILIA IBRU UNIVERSITY
AGBARHA-OTOR, DELTA STATE, NIGERIA
CSC 302: COMPILER CONSTRUCTION I

Instructor: OBIENU, Anayochukwu Chukwunonso, **Email:** obienuac@gmail.com

Lectures: Monday, 11am – 1pm; Tuesday, 10am – 11 am, LT1, **Phone:** (+234) 8037313900

Office hours: Mon, Wed, Fri, 1.30pm to 3.30 PM, **Office:**Power of Faith Building Floor 2 Rm 59

CSC 302: COMPILER CONSTRUCTION I (3 Units)

Course Category : Computer Sciences
Course Level : Undergraduate
Current Session : 2017/2018
Level : 300
Semester : 2nd

COURSE OBJECTIVES

At the end of the course, you should be able to:

- I. The basic theory of compilers.
- II. Evaluate the issues involved in implementing a compiler
- III. Good knowledge about how a compiler works

PREREQUISITE(S)

Csc 302 requires basic understanding of at least one programming language such as C, C++ etc. It would be an additional advantage if you have had prior exposure to Assembly Programming, Algorithm, Formal Languages and Automata.

COURSE DESCRIPTION

The course introduces the students to the operation of a modern compiler that translates code in a programming language into machine code. Special emphasis is placed on the actual construction of a compiler. Modern compilers operate in three passes. These are front-end, middle end and back end. Given a program written in a programming language, the front end carries out syntax analysis of input program. The middle end performs code optimization so that the eventual executable program will require less memory and will be efficient at runtime. The back end generates actual assembly language code for the target machine. Due to the introductory nature of the course, we will spend majority of the time on lexical analysis, syntax analysis and Parsers. The course provides an opportunity to see how concepts and techniques from various areas of Computer Science come together to build a useful tool.

COURSE OUTLINE

Review of Language Processors (Compilers, Assemblers and Interpreters). Phases of a Compiler. Structure and Functional aspects of a typical compiler, Lexical Analysis, Specification of Tokens, Recognition of Tokens, Top-down Parsing, Predictive Parsing, Recursive Descent Parsing, LL Parsing, LL Parsing Table Construction, Left Factoring, Top-down Parsing, Bottom-up Parsing, Shift-Reduce Parsing, Parser Generators, Semantic Analysis, Attribute Grammars, Ad-hoc scheme for attribute grammars, Intermediate Representation (IR), Intermediate Representation taxonomy, Error detection and recovery.

ASSESSMENT

The letter grades will be assigned using the following scale: A[70-100], B[60-69), C[50-59), D[45-49), and F[0-44).

Students will be assessed based on the following criteria:

- 75% Attendance of lectures,
- Class quizzes at the end of every course topic,
- Take home assignment
- Class Test
- Examination.

NB: Academic dishonesty will be "rewarded" with a grade of "F". "Sharing/reuse" of solutions to assignment problems is strictly prohibited.

RECOMMENDED BOOKS

1. *Compilers: Principles, Techniques, and Tools* (2nd Ed.), Aho, Lam, Sethi and Ullman, Addison Wesley, 2007.
2. *Compiler Construction: Principles and Practice*, Kenneth C. Loudon, PWS Publishing Company, 1997.

In addition to the above, the students will be provided with handouts by the lecturer.