Course Outline

1. Document Information

Degree Program	Computer Science
Course Number	CS 416
Course Title	Compiler Construction
Semester Hours	3
Course Coordinator	Khaled Ahmed
Revision Term	Spring 2020
Latest Revision	Fall 2020

2. Catalog Description

Introduction to compiler construction. Design of a simple complete compiler, including lexical analysis, syntactical analysis, type checking, and code generation.

3. Textbooks

- Aho, A., Sethi, R., & Ullman, J. (2007). Compilers: Principles, Techniques and Tools. Addison-Wesley, 2nd Edition. ISBN: 9780321486813.

4. References

- Tremblay, J. P. & Sorenson, P. G. (1985). The Theory and Practice of Compiler Writing. McGraw-Hill.

5. Course Learning Outcomes

- To learn the principles of compiler design and implementation.

6. Assessment of the Contribution to Student Outcomes

Outcome	1	2	3	4	5	6
Assessed		X	X			X

7. Prerequisites by Topic

CS 306 and 311 with a grade of C or better or graduate standing.

8. Major Topics Covered in the Course

1. Basic ideas: phases of a compiler, compiler construction tools {2 classes}
2. Language and grammars: basic concepts, classification of grammars (type 0, 1, 2, and 3), reduced grammars and extended BNF notations, regular expressions {4 classes}
3. A simple one-pass compiler: syntax definition, scanner, parsing, syntax directed translation, symbol tables, semantics and code generation {3 classes}
4. Lexical analysis: regular expressions, finite state acceptors, conversion algorithms, token specification, scanner generator (LEX) {6 classes}
5. Syntax analysis: top down parsing, recursive descent and predictive parsers, LL(1) grammars, bottom-up parsing, simple and operator precedence grammars, simple LR parsing, introduction to LALR and canonical LR parsing {6 classes}
6. Type checking: a simple type checker, type conversions {3 classes}
7. Symbol tables: symbol table organization for both block structured and non block structured languages {3 classes}
8. Run-time storage organization: dynamic storage allocation strategies, access to nonlocal names, parameter passing, heap storage {4 classes}
9. Intermediate codes: intermediate languages, quadruples {3 classes}
10. Code generation: issues in code design, target machine, register allocation, simple code generator {6 classes}