**CSC 326: Compiler Construction.**

**Pre-requisites:**

* CSC211 Data Structures and Algorithms,
* CSC213 Computer Architecture,
* CSC216 Assembly Language Programming,
* CSC222 Automata Theory,
* CSC223 Operating Systems.

**Learning Outcomes:**

* To understand the principles and techniques used to perform translation and the fundamental concepts of translator construction.
* To appreciate the concepts of compiler construction.
* To describe and analyze software tools and techniques which are applicable both to compilers and the implementation of system utility routines, command interpreters, etc.
* To construct simple compilers.

**Content:**

Compilers and Interpreters.

Overview of the compilation process: The phases of compilation:

* Lexical analysis, syntax analysis, semantic analysis, code generation;

Issues in compiler design: symbol tables, program compilation, loading and execution;

Compilation techniques:

* One pass and two pass. Run-time storage management. Object code for subscripted variables;

A simple complete compiler:

* Organization, Subroutine and function compilation.

Bootstrapping techniques,

* Multi pass compilation; Optimization: techniques, local, expressions, loops and global Optimization.

Software tools for compiler construction:

* Lexical analyzers; parser generators.

**Delivery:**

* lectures,
* Tutorials and Labs.

**Reading materials**

1. **Compilers**: Principles, Techniques, & Tools Second Edition by Alfred V. Aho, Columbia University; Monica S. Lam (Stanford University), Ravi Sethi Avaya, Jeffrey D. Ullman (Stanford University).
2. **Basics of Compiler Design,** Anniversary edition **by** Torben Ægidius Mogensen