

City University
Department of Computer Science & Engineering
Faculty of Science & Engineering

Course Outline

Course Code & Title: CSE 323, Compiler Design

Credit hours: 3

Prerequisites: CSE 319

Semester: Summer 2022

Total Weeks: 13

Hours/Week: 3

Total Hours: 39⁺

Instructor: Md. Touhidul Islam

Designation: Lecturer

Phone No: 01988892076

Office hours: By appointment

Email: touhid000cse@gmail.com

Rationale:

This course introduces students to fundamental principles and techniques in the design and implementation of modern programming languages. The course covers these issues mainly as they relate to functional programming and object-oriented programming. The emphasis is on rigorous mathematical formalism and reasoning, both in describing programming languages (their syntax) and in analyzing their behavior (their semantics).

Course Objectives:

- To **acquire** knowledge of the structure and the role of compilers and the principles and techniques used in the compilation process.
- To **cover** the methodologies learnt by the course to the development of small software projects regarding the construction of parts of compiler.
- To **assess** the features of the lexical and syntactic components of a syntax-directed translation specification and choose the most suitable solution for its development.
- To **understand** and explain the relationships between tools and results from the formal language theory and their application to the analysis phases of compilers.
- To **demonstrate** capacity for reading and understanding compiler internal representation for different kinds of compiler tasks.

Intended learning outcomes (ILOs) of the Course:

Knowledge	ILO 1: Will gain a thorough knowledge about the structure and the role of compilers and the principles and techniques used in the compilation process.
	ILO 2 : Will be able to understand the phases of the development of small software projects regarding the construction of parts of compiler.
	ILO 3: Will be able to evaluate the features of the lexical and syntactic components of a syntax-directed translation specification and choose the most suitable solution for its development to improve its performance.
	ILO 4: Will be able to analyze the major control flow properties of a program and expose the appropriate compiler internal representation for different kinds of compiler tasks..
Skill	Will develop the skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas.

Attitude	Will able to use regular expression and finite automata to design new language.
	Will develop skill to analyze written program syntactically and semantically.
	Develop attitude to group dynamics and team work.
	Build attitude to tackle challenges related to imaging problems.
	Create positive attitude to listen ideas of classmates.

Teaching Learning strategy:

- Classroom lectures, discussions, and review of theoretical concepts. Laboratory practical sessions.
- Class will be in a participatory mode.
- Office hours: Students are encouraged to make full use of the office hours of their instructor, where they can ask questions and go over lecture material.
- Use teaching formats such as discussion groups that encourage the participation of all students and help identify areas where students are having difficulties.
- Significant discussions and two-way communication are also expected during lectures to enrich the learning experience.
- Through problem solving and discussions in a small class room setting, tutorials provide opportunities for obtaining feedback. So provide learning materials in different formats (written, online, Power point slides etc) to support key concepts/knowledge. Particularly at the start of a program/module or for key areas, providing online or hard copy notes before classes can aid comprehension and accessibility.
- Include inquiry based learning exercises in course related contexts.
- Include group work, with groups representing diverse strategies.

Assessment Schedule

Assessment 1	Quizzes	Week 5 and Week 11
Assessment 2	Assignment	Week 3 and Week 9
Assessment 3	Presentation	Week 10
Assessment 4	Mid-Term Examination	As per university schedule
Assessment 5	Final Examination	As per university schedule

Assessment Strategy

Basis of Evaluation of Participants:	
Class Attendance	05%
Class performance including behavior	05%
Class Test	10%
Class Quiz/Assignment	10%
Midterm Examination	30%
Final Exam	40.0 %
Total	100.0 %

Course Outcomes:

After completion of this course each student will implement a compiler for a small programming language.

Text (Required):

A.V.Aho, Ravi Sethi, J.D.Ullman, Compiler Tools Techniques, Second Edition, Addison Wesley

Reference Books

- Trembley J.P. And Sorenson P.G.,The Theory And Practice Of Compiler Writing, First Edition, McGrawHill Robert
- V. Raghavan,Principles of Compiler Design, First Edition, McGrawHill
- Dick Grune, Henri E. Bal, Jacob, Langendoen,Modern Compiler Design, SecondEdition, WILEY India

Course Policies and Procedures:

General policies are as described in the university Bulletin.

All classes will be in a participatory mode.

To be eligible to sit for the Final Exam a student has to have minimum 80% class attendance. Attendance of anyone who will enter the class after the attendance register is closed may not be recorded. Any special consideration/provision for a student will be tied up absolutely with regularity in class attendance over the entire semester.

As and when needed, extra classes will be arranged to make up (possible) slow progress of the course. These classes will be considered as regular — attendance will be counted for these classes. Decisions regarding the schedules for these classes will be made in the class room when most of the students remain present in the class. No separate formal notice will be served for this purpose.

Students must have their own books and a separate note book exclusively meant for this course.

Random Quizzes:

All quizzes will be conducted on random basis. Random quizzes will not be made up other than for a student who is evaluated to be regular (100% attendance) before the day of the quiz in question and who can produce the evidence that he/she had the reason(s) for missing the quiz.

Assignments:

The details of assignments will be mentioned in the class as and when appropriate. No late assignments will be accepted. No erroneous or incomplete home assignments will be received.

Every assignment must have the following information at the top of the first page of the assignment:

ID#:
Name:

Assignment #:
Date assigned:
Due date:

Any assignment not conforming to this format will be subjected to random penalty. Assignments are to be submitted in 'letter size' or 'A4 size' papers.

Course Contents

Week	Topic	Reference Readings	Assignments	Due Date
1,2	Language translators: compilers and interpreters. The structure of a compiler: lexical analysis, parsing, semantic analysis, intermediate code generation, register allocation, global optimization. Bootstrapping a compiler.			
3,4	A Simple One-Pass Compiler: Syntax Definition, Syntax directed Translation, Parsing, Lexical Analysis, Symbol Table, Stack Machine.		Assignment 1	
5,6	Lexical scanning: Token classes, keyword recognition, minimizing the code-per-character cost of scanning, scanning numeric literals and string literals The interface between the scanner and the parser. Hand-written vs. automatically generated scanners.		Quiz 1, Presentation	
7	Midterm Exam			
8, 9	Semantic analysis: attributes and their computation, tree-traversals, visibility and name resolution. Inherited attributes and symbol tables. Name resolution in block-structured languages.			
10,11	Type checking. Type systems, varieties of strong typing, overload resolution, polymorphism and dynamic dispatching. Type-checking and type inference, unification.		Assignment 2	
11,12	Run-time organization: storage allocation, non-local references, parameter passing, dynamic storage allocation. Exception handling, debugging information. Intermediate code generation: control structures, expressions, simple register allocation. Aggregates and other high-level constructs.		Quiz 2	
	Final Term Exam			