Course Objective and Outcome Form
Department of Electrical and Computer Engineering
School of Engineering and Physical Sciences
North South University, Bashundhara, Dhaka-1229, Bangladesh

1. **Course Number and Title:** CSE326/426 Compiler Construction

2. **Number of Credits:** 3 Cr

3. **Type:** Elective, Science & Engineering, Lecture

4. **Prerequisites:** CSE 332 Computer Organization and Architecture

5. **Contact Hours:** 3 hours (theory)

6. **Instructor:** Dr. Rajesh Palit (RjP)
   Office: SAC 934
   Phone: NSU Ext 1508
   Email: rajesh.palit@northsouth.edu
   Office Hour: ST 1:30 PM – 2:30 PM
   Phone: +880 17 1955 7447

7. **Class Time, Room:** MW 11:20 AM – 12:50 PM, SAC 311

8. **Course Summary:** This course introduces the concepts of compilation of computer programs, including lexical analysis, context-free grammar, syntax analysis (parsing), syntax-directed translation, abstract syntax trees, types and type checking, intermediate languages, data flow analysis, program optimization, code generation, and runtime systems. Students will learn how a program written in a high-level language is systematically translated into a target program suitable for machines.

9. **Course Objectives:** The objectives of this course are,

   a. to introduce the phases of compilation process and compiler design
   b. to use finite automata to describe patterns for tokens for a given language and to define grammar for a programming language
   c. to introduce top-down and bottom-up parsing techniques such as predictive parsing and shift-reduce parsing and various ways to manipulate grammars
   d. to understand the role of semantic analyzer, type checking and intermediate code generation and to discuss various compiler optimization techniques that are applied before code generation
   e. to use compiler construction tools such as lex and yacc for generating scanners and parsers

10. **Course Outcomes (COs):**

    Upon Successful completion of this course, students will be able to:

Sl.	**CO Description**	**Weightage (%)**
1.	**Identify** the phases of a typical compiler, including the frontend and backend; **recognize** tokens of a typical high-level programming language; **describe** regular expressions for tokens and design;	15
2.	**Explain** the role of a parser in a compiler and **relate** the yield of a parse tree to a grammar derivation; **apply** an algorithm for a top-down	50

	or a bottom-up parser construction; **classify** different types of grammars.	
3.	**Explain** the role of a semantic analyzer, type checking and intermediate code generation process and **apply** the concepts for a given input string with code optimization;	20
4.	**Use** *lex* and *yacc* **tools** for building scanners and parsers for a given grammar; implement a lexical/syntax analyzer for a given grammar;	15

11. **Resources**

No	Name of Author(s)	Year of Publication	Title of Book	Edition	Publisher's Name	ISBN
1	Alfred V. **Aho**, Monica S. Lam, Ravi Sethi and Jeffry D. **Ulman**	2006	Compilers Principles, Techniques, & Tools	2nd	Pearson	ISBN: 978-81-317-2101-8
2	Doug Brown, John R. Levine, and Tony Mason	1992	Lex & Yacc	2nd	O'Reilly Media, Inc.	ISBN-13: 978-15-659-2000-2

Online resources: YouTube Tutorials, Google Classroom

12. **Weightage Distribution among Assessment Tools (Tentative)**

Theory		Lab	
Assessment Tools	**Weightage (%)**	**Assessment Tools**	**Weightage (%)**
Class Performance	5		
Quizzes	25		
Assignments & Viva	10		
Midterm	30		
Final Exam	30		

13. **Grading policy:** As per NSU grading policy available in

http://www.northsouth.edu/academic/grading-policy.html