National University of Sciences & Technology (NUST)
School of Electrical Engineering and Computer Science (SEECS)
Department of Computing

# CS-354: Compiler Construction

| | | | |
|---|---|---|---|
| **Course Code:** | CS-354 | **Semester:** | 7th |
| **Credit Hours:** | 3+1 | **Prerequisite Codes:** | CS-352 and CS-250 or Equivalent |
| **Instructor:** | Dr. Nasim Abbas | **Class:** | BSCS 7 (A & B) |
| **E-mail** | abbas.nasim@siat.ac.cn | **Knowledge Group:** | Core Computer Science |
| **Lecture days** | Tuesday & Friday | **Lab** | Thursday |

## Course Description:

CS-354 is an introduction to compiler design principles, tools and techniques. The topics covered include lexical analysis, syntax analysis, syntax directed translation, intermediate code generation and target code generation. Students are expected to be familiar with C/C++, Data Structures, Theory of Automata and Formal Languages and must be very comfortable with their programming techniques.

Compilers are one of the most important and commonly used tools for translating high-level software to machine code, including optimizations to generate high performance object code. Compilation systems are composed of many components and passes to ease their development, portability and scalability.

## Course Objectives:

The successful completion of this course would enable students to develop an understanding of compiler construction concepts, tools and key technique. Students should be able to:

- Understand the fundamental concepts, components and phases of a compilation system.
- Develop a prototype compiler for an arbitrary language or write a compilation pass for an existing compilation system such as GCC or LLVM.
- Look at compilation systems and see the different trade-offs such as power consumption, performance (compilation time/execution time/object code size) and code optimization.

## Course Learning Outcomes (CLOs):

| At the end of the course the students will be able to: | PLO | BT Level*** |
|---|---|---|
| 1. Design and implement scanners using a high level language and scanner generation tools. | C | C-3 |
| 2. Design and implement parsers using top-down and/or bottom-up paradigms. | B | C-3 |
| 3. Relate high-level code with the intermediate representations including Three Address Code (TAC) and Directed Acyclic Graphs (DAGs). | J | C-2 |
| 4. Explain code generation concepts including Basic Blocks, Control Flow Graphs and Register Allocation. | A | C-2 |

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

## Mapping of CLOs to Program Learning Outcomes

| **PLOs/CLOs** | **CLO1** | **CLO2** | **CLO3** | **CLO4** |
|---|---|---|---|---|
| PLO A (Computing Knowledge) | | | | X |

| | | | | |
|---|---|---|---|---|
| PLO B (Problem Analysis) | | X | | |
| PLO C (Design/Development of Solutions) | X | | | |
| PLO D (Individual and Team Work) | | | | |
| PLO E (Ethics) | | | | |
| PLO F (Communication) | | | | |
| PLO G (Societal Impact) | | | | |
| PLO H (Lifelong Learning) | | | | |
| PLO I (Modern tool usage) | | | | |
| PLO J (Investigation) | | | X | |
| PLO K (Project Management) | | | | |

**Mapping of CLOs to Assessment Modules and Weightages (In accordance with NUST statutes)**

To be filled in at the end of the course.

| **Assessments/CLOs** | **CLO1** | **CLO2** | **CLO3** | **CLO4** |
|---|---|---|---|---|
| Quizzes: 10% | | | | |
| Assignments: 5% | | | | |
| OHT-1: 12.5% | | | | |
| OHT-2: 12.5% | | | | |
| Labs: 25% | | | | |
| End Semester Exam: 35% | | | | |
| Total : 100 % | | | | |

**Marks Distribution (Tentative)**

**Theory**
Final: 40%
OHTs: 30%
Quizzes: 10%
Assignments : 10%
Research work: 10%

**Practical**
Lab work: 75%
Lab viva : 25%

**Books:**

| | |
|---|---|
| **Text Book:** | 1.Alfred V. Aho et al., **C**ompilers: **P**rinciples, **T**echniques and **T**ools, 2/E, Pearson, 2007 |
| **Reference Books:** | 1.Louden, Kenneth C. "Compiler construction." Cengage Learning .2012 |

| Week | Lecture Topic | Reading List |
|---|---|---|
| 01 | **Compilers:** Introduction of compilers, Structure and Applications | CPTT (§1.1 - §1.5) |
| 02 | **Lexical Analysis:** White Space, Looking Ahead, Keywords, Identifiers, Symbol Tables | CPTT (§2.6 - §2.7) |
| 03 | **Lexical Analysis:** Tokens, Patterns, Lexemes and Input Buffering | CPTT (§3.1-§3.2) |
| 04 | **Lexical Analysis:** Specification and Recognition of Tokens, Lexical-Analyzer Generator (Lex/Flex), Design of a Lexical-Analyzer Generator | CPTT (§3.3-§3.5, §3.8) |
| 05 | **Syntax Analysis:** Context Free Grammars, Parse Trees, Derivations, Precedence and Ambiguity | CPTT (§2.2, §4.2) |
| **06** | **Syntax Analysis:** Top Down (LL) vs. Bottom Up parsing (LR), Left Recursion, Predictive Parsing, Left Factoring | CPTT (§4.3-§4.5) |
| 07 | **OHT-1** | |
| 08 | **Syntax Analysis:** Recursive-Descent Parsing, Table-Driven Parsing, FIRST & FOLLOW Sets | CPTT (§4.4) |
| 09 | **Syntax Analysis:** LL(1) Grammars, LL(1) Parse Table Construction, LR Parsing, Handles, Handle-Pruning | CPTT (§4.4-§4.5) |
| 10 | **Syntax Analysis:** Shift-Reduce Parsing, Viable Prefixes, LR(k) Items, SLR(1), LR(1) and LALR Parsing | CPTT (§4.6-§4.7) |
| 11 | **Syntax Directed Translation:** Syntax-Directed Definitions, Synthesized and Inherited Attributes, Evaluation Orders, Applications of SDDs | CPTT (§5.1-§5.3) |
| **12** | **Intermediate Code Generation:** Directed Acyclic Graphs, Three Address Code, Types and Declarations | CPTT (§6.1-§6.3) |
| 13 | **OHT-2** | |
| 14 | **Intermediate Code Generation:** Translation of Expressions, Type Checking and Control Flow | CPTT (§6.4-§6.6) |
| 15 | **Code Generation:** Key Issues, Target Language and Addresses in Target Code, Basic Blocks, Flow Graphs | CPTT (§8.1-§8.4) |
| 16 | **Code Optimization:** Peephole Optimization, Register Allocation and Instruction Selection | CPTT (§8.7-§8.9) |

| **Lab Experiments** |
|---|
| 01 Introduction |
| 02 Lexical Analysis using C/C++ – Part I |
| 03 Lexical Analysis using C/C++ – Part II |
| 04 Lexical Analysis using Lex/Flex – Part I |
| 05 Lexical Analysis using Lex/Flex – Part II |

| |
|---|
| 06 Syntax Analysis using Bison – Part I |
| 07 Syntax Analysis using Bison – Part II |
| 08 Syntax Analysis using C/C++ – Recursive Descent Parsing |
| 09 Introduction to Frances – Part I |
| 10 Introduction to Frances – Part II |
| 11 LLVM – Compiling and Verifying Optimizations using LLVM |
| 12 Compiler Optimization |
| 13 Compiler project problem solving |

| **Tools / Software Requirement:** |
|---|
| Visual Studio C++ , Linux (Ubuntu 14.04 LTS), flex v2.5+, bison v2.7+, Raspberry Pi (Raspbian operating system) |

| **Grading Policy:** | |
|---|---|
| **Quiz Policy:** | The quizzes will be announced and normally last for fifteen minutes. The question framed is to test the concepts involved in last few lectures. Missed quizzes will not be compensated. |
| **Assignment Policy:** | The course website (LMS portal) will be the primary source for announcements and submitting assignments. All homework assignments must be done individually or as directed<br>May also require uploading a soft-copy on LMS (for plagiarism check).<br>**Late Submissions**<br>Late submission will get a –20% penalty up to one day<br>Contact instructor at least 03 days before the deadline, if case of emergency |
| **Lab Conduct:** | The labs will be conducted for three hours every week. The lab handouts will also be placed on LMS. The students are to submit their lab tasks at the end of lab for evaluation. One submission per group will be required. However, students may also be evaluated by oral viva during the lab. |
| **Plagiarism:** | Your writings must be your own thoughts. You must cite and acknowledge all sources of information in your assignments. Cheating and plagiarism will not be tolerated and will lead to strict penalties including zero marks in assignments as well as referral to the Dean for appropriate action(s). |

**Dr. Nasim Abbas Profile**

I received PhD degree in Pattern Recognition & Intelligent Systems (Specialization: Congestion control for wireless multimedia sensor networks) from Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China. Chinese Academy of Sciences is ranked 1$^{st}$ in world in terms of research. During my PhD, I published 8 papers as first author. My papers are published in 3 SCI journals and 5 IEEE conferences (including 1 class “A” conference (in India) and 1 class “B” conference (in USA)). Moreover, during my PhD, I was awarded combined scholarship of Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, and Shenzhen government foundation. Furthermore, I received my MS from Muhammad Ali Jinnah University, Islamabad, and BS from COMSATS Institute of Information Technology, Islamabad. My research interests include wireless communication, wireless ad-hoc networks, and wireless multimedia wireless sensor networks.