INTERNATIONAL UNIVERSITY VNUHCM

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

IT092IU – PRINCIPLES OF PROGRAMMING LANGUAGES

PROJECT ASSIGNMENT

Due date: 11:00 PM, Saturday, 22 May 2021

Weighting: **15%** of the final score

Objectives

This assessment task addresses all objectives in the subject outline.

Overview

In this assignment, you will

- Stimulate how a compiler works with a programming language
- Learn how to apply at least 1 programming language in implementing a system
- Work on the project as a team

SPECIFICATION

Your team is required to develop a compiler simulator, named *"COSIM",* for a programming language of your own choice (e.g., Java, C/C++, Python, etc.) which must be clearly stated early in your documentation. This is the inputted language of your COSIM system, called *Language X*.

To develop COMSIM, your team is required to use Python as a primary programming language, and optionally use one or more other programming languages of your own interest.

INPUT: Your software receives a user-provided program whose source code uses the Language X and has at least 2 functions:

- 1 user-defined function, and
- 1 main function that uses the above user-defined function.

The program with these 2 functions follows the below requirements:

- **The user-defined function** performs mathematical calculation, including basic operations (+, -, *, /, absolute value, sin, cos, tan, power) and at least 1 constant (E, PI or any user-defined constant).

- **The main function** uses the previously user-defined function, performs sequence, if-then-else branching and basic input/output functions (to receive input or output the results on computer screens, such as *cin/cout, printf, System.out.println,* etc.) Optionally, your program can consider any advanced construct, such as loops (while-do, do-while, …), switch-case branching, etc. Clearly describe such advanced features in your document (if any).
- The program must work on its relevant (real) compiler and your COSIM. These functions can be considered and treated as *the character stream* of Figure 1.

OUTPUT: There are 2 output types:

1. Output of the **execution results** of the source code in Language X (including syntax errors and other possible errors).
2. Output of the behind-the-scenes execution of the simulator: presenting the **symbol table** and the **results of the first 3 phases** in the *phases of compilation* as described in Figure 1:
   a. scanner (lexical analysis),
   b. parser (syntax analysis),
   c. semantic analysis and intermediate code generation.

The output should be visually presented to reflect the nature of the results (e.g., a table for the symbol table, a tree for the syntax tree, etc.)

DELIVERABLES

1. **Report**
   a. A report (in .docx) must be submitted.
   b. The report must include a cover page that clearly specify all the team members' ID, Full name, Contribution Percentage %. Scores of all members might vary due to his/her contribution made to the project.
   c. Detailed solutions to the problems stated above.
   d. Screenshots should be included to clearly illustrate your results

   e. Lessons Learned should be added to present what your team has learned in this project.

2. **Source code**, including 2 demo programs of Language X (each program has 2 functions above).

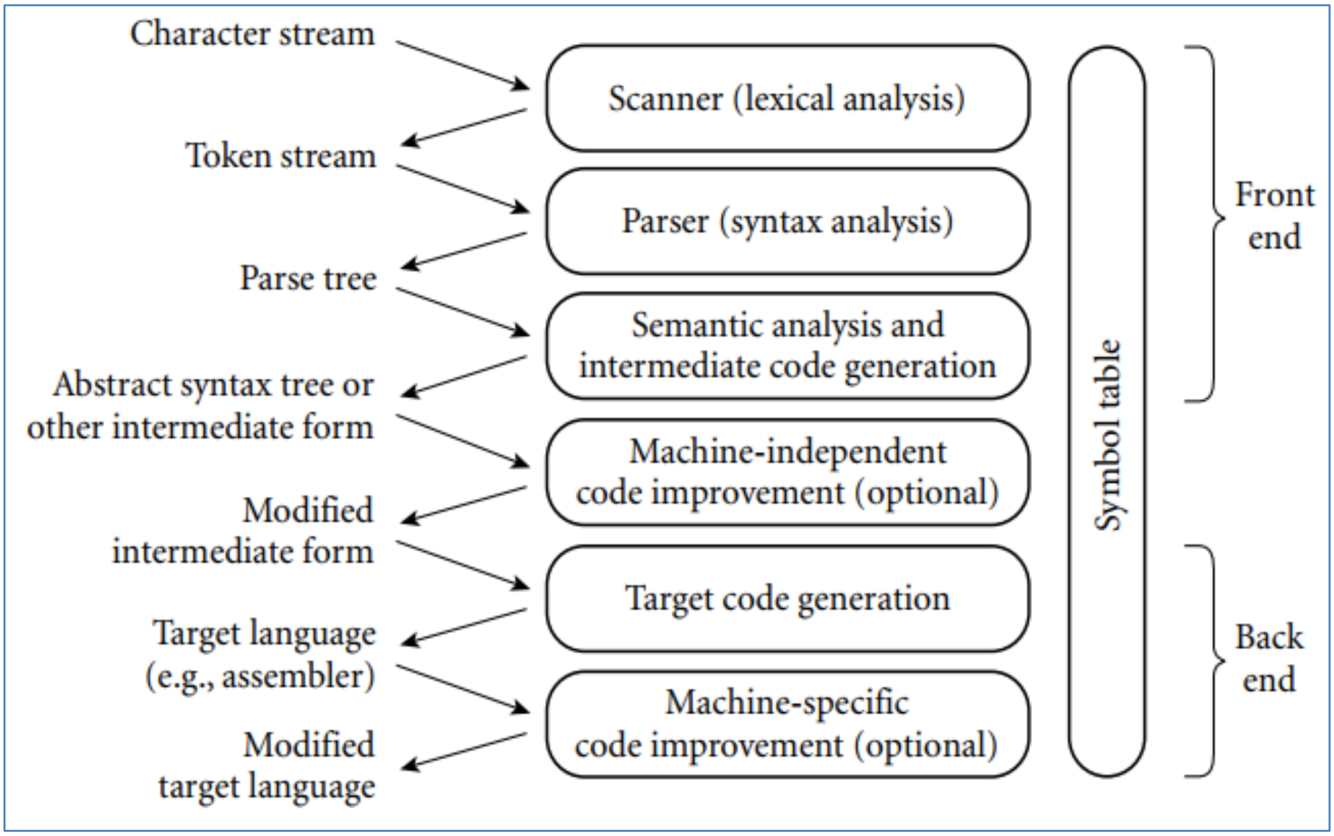

Character stream
Scanner (lexical analysis)
Token stream
Parser (syntax analysis)
Parse tree
Semantic analysis and intermediate code generation
Abstract syntax tree or other intermediate form
Machine-independent code improvement (optional)
Modified intermediate form
Target code generation
Target language (e.g., assembler)
Machine-specific code improvement (optional)
Modified target language
Symbol table
Front end
Back end

Figure 1. Phases of Compilation (the textbook [1] by Scott 2016, p. 50)

SUBMISSION

Your deliverables must be **submitted via email** to all of the following:

1. The Instructor Nguyen Hong Quang: iu.subjects@gmail.com
2. The Lab Instructor Nguyen Quang Phu: nqphu@hcmiu.edu.vn
3. **ALL** of your team members (the purpose of this submission email is to acknowledge their contribution percentage to the project).

LATE SUBMISSION

Late submission is allowed for up to 1 day late, which will incure 5% reduction of your project score.

DEMO

Each team will do a demo of your software during your lab's time during the week **between 24 and 29 May 2021**. Your team's demo session **must be registered with the Lab Instructor**: **Mr. Nguyen Quang Phu.**

MARKING SCHEME

NO.	CRITERIA	MAX POSSIBLE SCORE (POINTS)
1	Executable and workable software as described in the specification, and well-documented report	20
2	Results of Scanner	15
3	Results of Parser	15
4	Results of Semantic Analysis (and optionally intermediate code generation)	15
5	Results of Symbol Table	15
6	Source code and the 2 demo programs	10
7	Lessons Learned	5
8	Demo (team work, explanation…)	5
9	*Bonus for a comprehensive solution*	*10*
	Total max score:	**100**

Note: The maximum score of this project (including the bonus) is: 100 points.

APPENDIX 1. MORE ABOUT THE PHASES OF COMPILATION

Figures 2 and 3 present more pictures of the compilation process for your reference.

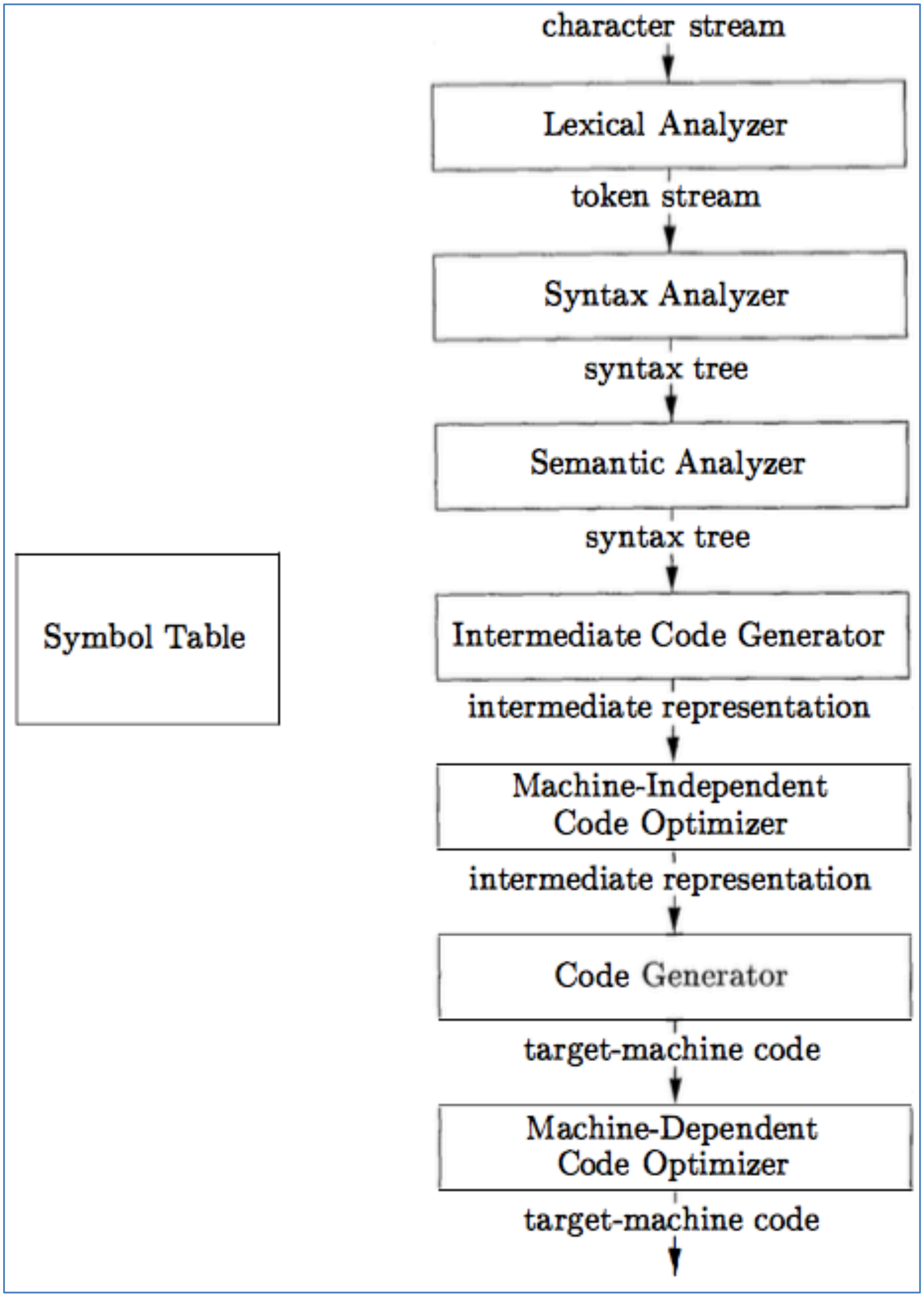
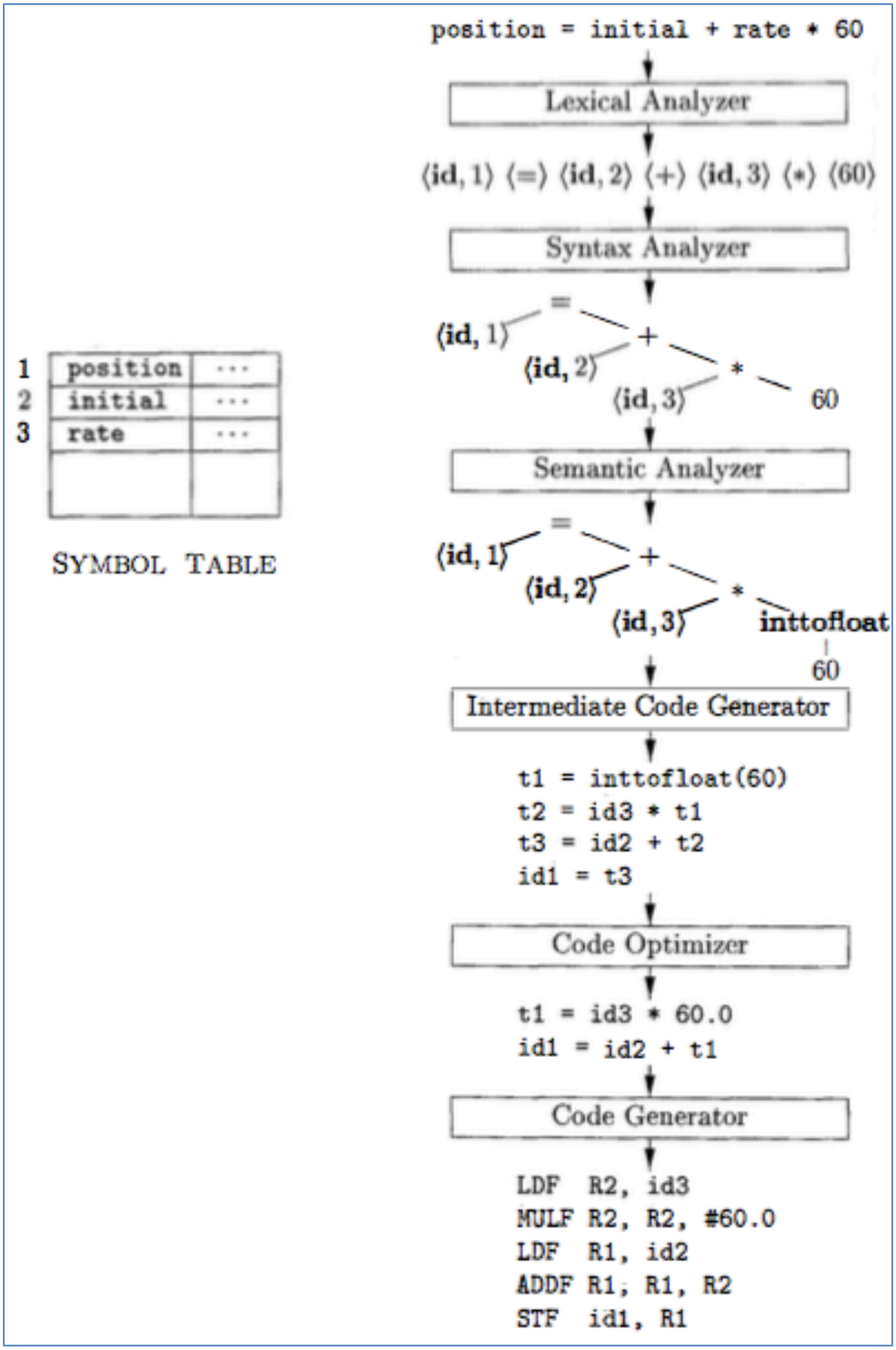


Figure 2. Phases of a compiler (Aho et. al, 2016)



```
position = initial + rate * 60
```

```
t1 = inttofloat(60)
t2 = id3 * t1
t3 = id2 + t2
id1 = t3
```

```
t1 = id3 * 60.0
id1 = id2 + t1
```

```
LDF  R2, id3
MULF R2, R2, #60.0
LDF  R1, id2
ADDF R1, R1, R2
STF  id1, R1
```

Figure 3. Example on translation of an assignment (Aho et. al., 2016)