XIAMEN UNIVERSITY MALAYSIA
厦門大學馬來西亞分校

OFFICE OF ACADEMIC AFFAIRS
Reference No. : XMUM.OAA - 100/2/8-V2.0
Effective Date : 23 APRIL 2018

DESCRIPTION OF COURSEWORK

Course Code	CST302
Course Name	Compiler Principles
Lecturer	Dr Mohammed Alswaitti / Dr Kamran Siddique
Academic Session	2020/09
Assessment Title	Lab Report

A. Introduction/ Situation/ Background Information

This report aims to drive the students to discover how practically the compiler works. The students will use Lex and Yacc to implement a miniature compiler (calculator) which gives them more insight into the compiler design process. Furthermore, it will enhance the affective part by presenting their work and highlight their additions to the given compiler

B. Course Learning Outcomes (CLO) covered

At the end of this assessment, students are able to:

CLO 4 Build a miniature compiler to translate a source language to a specified target language. (P4, PLO3).

CLO 5 Explain the types of compiler construction tools such as lexical and parser generators used in building compilers. (A4, PLO5).

C. University Policy on Academic Misconduct

1. Academic misconduct is a serious offense in Xiamen University Malaysia. It can be defined as any of the following:
   i. **Plagiarism** is submitting or presenting someone else’s work, words, ideas, data or information as your own intentionally or unintentionally. This includes incorporating published and unpublished material, whether in manuscript, printed or electronic form into your work without acknowledging the source (the person and the work).

ii. **Collusion** is two or more people collaborating on a piece of work (in part or whole) which is intended to be wholly individual and passed it off as own individual work.

iii. **Cheating** is an act of dishonesty or fraud in order to gain an unfair advantage in an assessment. This includes using or attempting to use, or assisting another to use materials that are prohibited or inappropriate, commissioning work from a third party, falsifying data, or breaching any examination rules.

2. All the assessment submitted must be the outcome of the student. Any form of academic misconduct is a serious offense which will be penalized by being given a zero mark for the entire assessment in question or part of the assessment in question. If there is more than one guilty party as in the case of collusion, both you and your collusion partner(s) will be subjected to the same penalty.

D. Instruction to Students

* This is a group report with 2 (at least) - 4 (at max) students per group.

* You are required to submit your report as a soft copy (Docx and pdf) and a short video link (Drive or any online storage) at **Moodle Lab_Report submission link**. All files should be **compressed as one file.Rar or Zip** (only one team member is enough for the submission). The deadline is by **8th January 2021**

* The following details that are required and should be written clearly on the **attached cover page**:

- Student Names
- Student ID numbers

* This report accounts for a total of 20 % of overall course evaluation.
* Attach the **marking rubrics to the end of your report.**
* Read the instructions carefully at Section F and any missing submissions will be penalized.

E. Evaluation Breakdown

No.	Component Title	Percentage (%)
1.	Report (Soft copy with source codes).	15
2.	A short video that presents and explains the work.	5
	TOTAL	**20**

F. Task(s)

Read the attached file (LexAndYacc.pdf) which contains a full description, steps, and details for building a calculator (which is considered as our small compiler). Then implement and code the associated Lex and Yacc definitions, functions, and any related issues. This is a self-learning process, you should learn how to configure Lex and Yacc to work with C.

To Turn In (What you have to submit)

You should turn in **your complete** code (. **Zip file**), a **full lab report** (**both .docx and .pdf**) that contains all your work, and a **short video link (any online storage service)** that explains and presents your work:

1. A detailed report that summarizes the implementation of the calculator in your own words starting from how we can download Lex/Yacc and ending with the integration of all parts and compilation. You can consider the report as a brief lab manual for teaching purposes, hence, the report should be easy, readable, and clear with snapshots of examples as an input and the corresponding outputs as well. Use your creativity to make this report rich with information, does not exceed 25-30 pages as a max, show what this calculator can do and can not do, etc...
2. The difficulties that you faced through the implementation of the calculator and how you have solved it (if applicable).
3. Recommendations of any better or simpler generators similar to Lex and Yacc that you found could be used (if applicable).
4. A short video that explains what you have done and how to run your code. You can consider it as a presentation session with restrictions on the length (7 mins max) and the size of the video (100 MB). All members should participate in the video. You can use any video recorders or compressors to get high quality with a small size.
5. Adding more features to the calculator will be counted as a bonus.

You should test your code. When you are confident that it is working correctly, send your submission to the Moodle (soft copy of the report, the code, and the video as one compressed file)

Note: You are free to choose another software or programming language, the main goal of this report is to learn how to use generators. Make your assumptions for any related matters and explain it if necessary.

APPENDIX 1

MARKING RUBRICS

Component Title			Percentage (%)		
Criteria	Score and Descriptors			Weight (%)	Marks
	Excellent (71-100 % of the weight)	Average (31-70 % of the weight)	Poor (0-30 % of the weight)		
Formating, consistency, readability, and clarity of the parts and their functions.	Complete, clear, and nearly error-free with all results	Contains some errors / missing definitions/ not informative parts	An incomplete or large amount of errors and lack of results	2	
The functionality of the code, samples of inputs, and outputs.	Complete and nearly error-free with all results	Contains some errors / missing output	An incomplete/ not executable/ large amount of errors and lack of results	10	
Some samples of Justifications of what works and what does not work and why.	Clear examples with reasonable justifications	Missing examples / non-convenient justifications	No examples/ poor examples and justifications	3	
The clarity and the simplicity of the presentation	Clear and meet the video restrictions	Have some ambiguity and vague parts	Lack of clarity and almost useless	5	
Additional features, suggestions, and comments	Persuasive and Convenient additions with evidence (not theoretical) will be given 1-3 bonus marks.			bonus	
			TOTAL	20	

Note to students: Please print out and attach this appendix together with the submission of coursework