CEBU INSTITUTE OF TECHNOLOGY – UNIVERSITY
COLLEGE OF COMPUTER STUDIES
COMPUTER SCIENCE DEPARTMENT

COURSE GUIDE

Course / Section	**CS313-Automata Theory and Formal Languages**	Credit Units	3
Instructor	CHERRY LYN STA. ROMANA (cstaromana@cit.edu)	Total Hours	54
Pre-Requisite/s	Design and Analysis of Algorithms		

Course Description

The course introduces some fundamental concepts in automata theory and formal languages including grammar, finite automaton, regular expression, formal languages, pushdown automaton, and Turing machines; as well as the applications of these theories in programming languages, compiler design and construction and software development in general.

This course is one of the most important courses in Computer Science. Theoretical foundations governing programming languages, compiler design, designing systems in general are covered in this course.

Course Learning Outcomes

CLO1: Explain the importance of the course in relation to the goals and objectives of the program, the college and the university
CLO2: Demonstrate knowledge of alphabets and languages
CLO3: Design/Construct finite-state machines, regular expressions, context-free grammar, push-down automata and Turing machines for modeling a given language
CLO4: Design and implement simple lexical analyzers and parser for simple languages using automata concepts
CLO5: Discuss the various applications of computer theory in software development.

Topics / Modules and Intended Learning Outcomes

Main Reference: *Elements of the Theory of Computation by Lewis and Papadimitriou*

Topic 1: Alphabets, Languages and Regular Expressions (pages 47-51)
ILO1. Define the terms symbol, alphabet and languages
ILO2. Determine whether a string is generated by a regular expression
ILO3. Construct regular expression to represent a particular language

Topic 2: Finite Automata (pages 55 - 67)
ILO1. Define a finite automaton
ILO2. Determine thru tracing whether a particular string is accepted by a DFA
ILO3. Construct a DFA to represent a particular language
ILO4. Determine thru tracing whether a particular string is accepted by a NFA
ILO5. Construct a NFA to represent a particular language

Topic 3: Equivalence of DFA, NFA and Regular Expressions (pages 67-85)
ILO1. Convert an NFA to a DFA
ILO2. Convert a regular expression to NFA
ILO3. Convert a DFA to a regular expression

ILO4. Explain and illustrate the relationship among regular expression, DFA and NFA

Topic 4: Applications and Properties of Finite Automata and Regular Languages (pages 86-90)

ILO1. Prove that languages are regular

ILO2. Prove that languages are not regular

ILO3. Apply the theory of finite automata in the construction of a lexical analyzer and other string processing applications

ILO4. Explain how finite automata may be used as a modeling tool in system development

Topic 5: Context-free Grammar (pages 113-120)

ILO1. Differentiate a regular language from a context-free language

ILO2. Define a context-free grammar (CFG)

ILO3. Determine whether a string is generated by a Context-free grammar

ILO4. Construct a context-free grammar for a particular language

ILO5. Construct a regular grammar for a regular language

Topic 6: Push-down Automata (pages 130-150)

ILO1. Define a push-down automaton (PDA)

ILO2. Determine whether a string is accepted by a push-down automaton

ILO3. Construct a push-down automaton for a particular language

ILO4. Prove by construction that a language is context-free

ILO5. Explain the relationship between CFG and PDA and regular languages

Topic 7: Turing Machines (pages 179 – 200)

ILO1. Construct a Turing machine to compute a function

ILO2. Determine whether a string is accepted by a Turing machine

ILO3. Construct a Turing machine to decide a language

ILO4. Construct a Turing machine to accept a language

ILO5. Prove by construction that a language is Turing decidable

ILO6. Prove by construction that a function is Turing computable

ILO7. Prove by construction that a language is Turing acceptable

MADE4Learners FRAMEWORK

Approach		Distance Blended		Distance		Online	/	**Approach 3**
Technical/Software Requirements	○ Hardware: PC / laptop ○ Software/App: PDF reader, MSTeams ○ Materials, etc: Wifi connection at home ○ MS Teams							

Class Guidelines

- I am expecting that you have the maturity and discipline to do self-directed learning.
- Integrity is also very important. I expect that you will answer assessment activities on your own without cheating or copying from your classmates.
- This course is highly mathematical. Thus, there are a lot of analysis and problem solving required. This course also builds on topics from the beginning to the end. It will be hard to acquire learning outcomes of succeeding topics if you have not acquired the learning outcomes of previous topics.
- Assessment activities will naturally be in-line with the learning outcomes. Do not ask me what type of questions because it will be aligned with the outcomes.
- The most important outcome is your ability to design/contruct models like DFA, NFA, CFG, PDA, TM etc. Naturally, the assessment will also require you to design/construct DFA, NFA, CFG, PDA, TM etc for modeling languages. To prepare for this, you need to practice.
- I will do my best to provide you with learning resources.

○ MS Teams will be our major official form of communication. Announcements and discussion threads will be posted there. It will also be used for file sharing of ebooks, notes, my video recordings.	
GRADING SYSTEM	
Midterm Grade	MCS (Midterm Class Standing) – activities, quizzes, machine problem PE – Prelim Exam ME – Comprehensive Midterm Exam MS (Midterm Score) = PE * 0.20 MCS * 0.40 + ME * 0.40
Final Grade	FCS (Final Class Standing) – activities, quizzes, machine problem PFE – Prefinal Exam FE - Comprehensive Final Exam FS (Final Score) = MS * 0.5 + FCS * 0.2 + FE * 0.2 + PFE * 0.10