Course Outline

A. Basic Information

Semester : Fall 2024

Course Code : **CSE441**

Course Title : **Theory of Computing**

Credit : 3.0

Pre-requisite Courses :

Course Code	Course Title
CSE265	Algorithm

Course Offering Department : Department of Computer Science and Engineering

Faculty : Nahid Hasan [NH]

Class Schedule :

Course Code	Section	Room Number	Day	Start Time	End Time

Consultation Schedule :

Day	Start Time	End Time	Duration

Contact Number : +880 1534561707

Email Address : hasan.nahid@seu.edu.bd

B. Routine of Faculty

Day	Class Hours			Counseling Hours Room: SEU606	
	Courses	Time	Room	Day	Time
Sunday				Sunday	
Monday				Monday	
Tuesday				Tuesday	
Wednesday				Wednesday	
Thursday				Thursday	
Friday and Saturday	Off Day				
To make appointment at some other time (except counseling hour), please contact via email					

C. Course Details

1. Importance of the Course:

It is a fundamental course that provides a deep understanding of the principles underlying computation, including its capabilities and limitations. It introduces abstract models like finite automata, Turing machines, and pushdown automata, which form the basis for analyzing algorithms and computational processes. This course is essential for studying computational complexity, helping to classify problems based on their solvability and efficiency. By bridging mathematics and computer science, it enables students to design efficient algorithms, understand the theoretical aspects of programming languages, and explore emerging fields like cryptography and machine learning. Overall, it lays the groundwork for advanced studies and practical applications in computer science.

2. Objectives

This course is intended to help students to:

1. Explore and analyze abstract models of computation, such as finite automata, pushdown automata, and Turing machines, and their role in understanding computational processes Understands the core concepts and mathematical foundations of computer graphics
2. Examine the boundaries of what can and cannot be solved by a computer, including undecidable problems and computational intractability. Demonstrate foundational knowledge of modeling and representations of 3D shapes.
3. Develop skills in designing and analyzing algorithms, focusing on time and space complexity, and understanding complexity classes like P, NP, and NP-complete.
4. Apply theoretical concepts to practical problems in areas such as programming languages, cryptography, and machine learning, fostering critical thinking and problem-solving skills.

3. Course Outcomes (COs)

At the end of the course, the students will be able to:

COs	Description	POs	Domain/Level of Learning Taxonomy	Teaching-Learning Strategy	Assessment Strategy
CO1	**Students will be able to explain the fundamental concepts of automata theory**	PO1	L1, L2	Lectures, Group Discussion, Presentation	**Formative:** - Class Test - Discussion - Problem-based Exercise - Viva **Summative:** - Mid-Term Exam
CO2	Students will apply formal methods to prove language properties, construct grammars, and determine the computational power of various abstract machines.	PO2	L2, L3,L4, L5	Lectures, Group Discussion, Presentation	**Formative:** - Class Test - Discussion - Problem-based Exercise - Case Study - Viva **Summative:** - Mid-Term Exam - Final Exam
CO3	Students will develop the ability to design and analyze different computational models, such as finite automata, pushdown automata, and Turing machines, for solving computational problems.	PO3	L2, L3,L4, L5	Lectures, Group Discussion, Presentation	**Formative:** - Class Test - Discussion - Problem-based Exercise - Case Study - Viva **Summative:** - Mid-Term Exam - Final Exam

CO4	**Students will be able to assess problems in terms of decidability, reducibility, and computational complexity, including P, NP, and NP-complete classes.**	PO4	L1, L2,L3	Lectures, Group Discussion, Presentation	**Formative:** - Class Test - Discussion - Problem-based Exercise - Case Study - Viva **Summative:** - Mid-Term Exam - Final Exam

4. Course Outcomes (COs) and Program Outcomes (POs) Mapping

CO	PO											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓											
CO2		✓										
CO3			✓									
CO4				✓								

5. Tentative Lecture Plan

Class	Lecture No.	Contents	Learning Outcome	Learning Resources	CO's
01	Lecture 1	Concept of theory of computing, compiler and interpreter , Symbol, Alphabet, String , Regular Language.	Understand the principles of computation, formal languages.	Lectures, Slides, Book	CO1
02				Lectures, Slides, Book	
03	Lecture 2	Properties of regular languages, minimizing automata, regular expressions.	Design and interpret regular expressions to represent patterns and generate equivalent finite automata.	Lectures, Slides, Book	**CO1**
04				Lectures, Slides, Book	
05	Lecture 3	Formal proofs, Context Free Grammar (CFG), Ambiguity in CFG.	Construct and analyze CFGs to generate and describe context-free languages.	Lectures, Slides, Book	**CO1,CO2**
06				Lectures, Slides, Book	
07	Lecture 4	Finite automata.	Design, analyze, and apply finite automata (DFA and NFA) to recognize and process regular languages effectively.	Lectures, Slides, Book	**CO1,CO3**
08				Lectures, Slides, Book	

09	Lecture 5	Deterministic Finite Automata (DFA).	Design, analyze, and apply finite automata (DFA and NFA) to recognize and process regular languages effectively.	Lectures, Slides, Book	CO1,CO3
10					
11					
12					
13	Lecture 6	Non Deterministic Finite Automata (NFA) , NFA to DFA.	Understand and construct NFAs to recognize regular languages using multiple possible transitions for a given input, Convert an NFA into an equivalent DFA using the subset construction method to ensure deterministic language recognition.	Lectures, Slides, Book	CO1,CO3
14				Lectures, Slides, Book	
15				Lectures, Slides, Book	
16	Lecture 7	Epsilon NFA to DFA.	Convert an NFA into an equivalent DFA using the subset construction method to ensure deterministic language recognition.	Lectures, Slides, Book	CO3
17				Lectures, Slides, Book	
18	Lecture 8	Pushdown Automata, Turing Machine.	Design and analyze PDAs to recognize and process context-free languages using a stack-based memory structure, Understand and construct Turing Machines to model computation and solve problems involving recursively enumerable languages.	Lectures, Slides, Book	CO1, CO3
19				Lectures, Slides, Book	
20				Lectures, Slides, Book	
21	Lecture 9	Pumping lemma, Non-regular languages, Decidable languages.	Identify and prove that certain languages are non-regular using the pumping lemma or other theoretical techniques.	Lectures, Slides, Book	CO4
22				Lectures, Slides, Book	
23		Undecidability of the Halting	Understand the principles	Lectures, Slides,	CO4

	Lecture 10	problem , Classes P, NP,NPC , Diagonalization and reduction .	of computation	Book	
24				Lectures, Slides, Book	

Teaching and Learning Methods

- Online Learning Management System (Google Classroom)
- Lecture delivery in Physical Class
- Lecture materials in Google Classroom
- Discussion during class and counseling hours
- Sample codes provided during physical class and via Google Classroom

6. Assessment

i.Tentative Assessment Schedule

Serial	Assessment Type	Schedule	Comments
1.	Quiz 1	Week 05	Syllabus and announcements will be given ahead of time.
2.	Assignment 1	Week 04	Problem set will be uploaded in the classroom.
3.	Midterm Exam	Week 07	Announcements will be given ahead of time.
4.	Quiz 2	Week 09	Syllabus and announcements will be given ahead of time.
5.	Final Exam	Week 14	Announcements will be given ahead of time.

ii. Tentative Weight Assessment

Assessment Tools	Percentage
Attendance	10%
Class Tests	15%
Assignments/Presentation	25%
Mid Term Examination	20%
Final Examination	30%
Total	**100%**

7. Grading Policy

Obtained Marks		Letter Grade	Grade Point	Assessments
Minimum	Maximum			
80%	100%	4.00	A+	Outstanding
75%	79%	3.75	A	Excellent
70%	74%	3.50	A-	Very Good
65%	69%	3.25	B+	Good
60%	64%	3.00	B	Average
55%	59%	2.75	B-	Below Average
50%	54%	2.50	C+	Poor
45%	49%	2.25	C	Very Poor
40%	44%	2.00	D	Passing
0%	39%	0.00	F	Fail

8. Lecture Materials

Lecture Notes	As provided during class
Text Book(s)	Compiler Principles, Techniques and Tools By Aho, Ullman, Sethi.
Reference Book(s)	Compiler construction, Principles and Practice, By Kenneth C Louden.
Online	Resources as provided during class time

Resources	

9. Aiding Materials for Learning

- Internet Connectivity
- SEU official email ID.
- Should know how to use “Google Meet” and “Google Classroom”

11. Faculty Suggestions

1. The date and syllabus of the lectures, class-tests, midterm, and final exam are already given here; however, announcements will be given ahead of time. There is **NO** provision for make-up class tests.
2. The reading materials for each class may be given before that class so that students may have a cursory look into the materials. All materials (lecture notes, supporting reading materials, etc) will be made available through google classroom (classroom.google.com).
3. Class participation is vital for a better understanding of the subject matter. Class will be conducted in an interactive environment where both the teacher and the students must pose questions and discuss solutions for better understanding.
4. Mobile phone or other devices **MUST** keep silence during the class and exam periods.
5. A student who cheats, plagiarizes, or furnishes false, misleading information in the course is subject to disciplinary action up to and including an F grade in the course and/or suspension/expulsion from the University.
6. Students must maintain the code of conduct specified by SEU.
7. The goal of any assignment is to give you practice in mastering the course material. Consequently, you are encouraged to collaborate on problem sets. In fact, students who form study groups generally do better on exams than do students who work alone.
8. You must write up each problem solution by yourself without assistance. It is a violation of this policy to submit a problem solution that you cannot explain verbally to the course teacher.
9. No collaboration whatsoever is permitted during examination.