AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH (AIUB)

Faculty of Science and Technology (FST)
Department of Computer Science (CS)
Undergraduate Program

COURSE PLAN

Spring 2020-2021 SEMESTER

I. Course Core and Title

CSC 2211: Algorithms

II. Credit

3 credit hours (2 hours theory and 3 hours Lab per week)

III. Nature

Core Course for CS, CSE, COE, CSSE, SE, CIS

IV. Prerequisite **CSC 2105**

V. Vision:

Our vision is to be the preeminent Department of Computer Science through creating recognized professionals who will provide innovative solutions by leveraging contemporary research methods and development techniques of computing that is in line with the national and global context.

VI. Mission:

The mission of the Department of Computer Science of AIUB is to educate students in a student-centric dynamic learning environment; to provide advanced facilities for conducting innovative research and development to meet the challenges of the modern era of computing, and to motivate them towards a life-long learning process.

VII - Course Description:

The design and analysis of algorithms is the core subject matter of Computer Science. Designing an algorithm for a computational problem involves knowledge of the problem domain, a thorough knowledge of the data structures that are available and suitable and no small measure of creativity. This course concentrates on the above problems, studying useful algorithmic design techniques, and methods for analyzing algorithms.

- Discuss the principles and objectives of Algorithms along with its purpose and necessity in the program domain.
- Distinguish computational problems with respect to inputs and outputs in addition to their solutions in efficient ways.
- Simplify in different ways to find out solutions of fundamental computational problems, their solutions and performances.
- Justify the necessary and sufficient condition behind a solution of any widely accepted or self-developed algorithm.
- Demonstrate well known algorithmic solutions of different problems as essential parts of study along with recent improvements.
- Discuss efficiencies of different types algorithms on different problem domains.
- Analyze time and space complexities of any widely accepted or self-developed algorithm.
- Apply appropriate data structures to implement the efficient algorithms.
- Explain classical tools and techniques for algorithms analysis and design.
- Judge creativity in designing algorithms.

VIII – Course outcomes (CO) Matrix:

By the end of this course, students should be able to:

CO	CO Definition	Blooms Level	Blooms Learning Level			PO Map	Assessment Method
			C	P	A		
CO1	Outline terminologies and methods of classical algorithms analysis and design to solve some conventional/real life algorithmic problems.	1	C			1	Quiz
CO2	Demonstrate the selective well known and self-developed (if any) algorithms in efficient ways.to solve some engineering problems.	3	C			1	Mid / Final Term
CO3	Break down some complex realistic problems into modular problems and their solutions.	4			A	2	Assignment 1
CO4	Justify the usage of the algorithms and their data structures to solve some complex realistic problems along with complexity analysis.	5			A	2	Assignment 2

C: Cognitive; P: Psychomotor; A: Affective; S: Soft-skills (CT: Critical Thinking, TS: Teamwork)
**The numbers under the 'Level of Domain' columns represent the level of Bloom's Taxonomy each CO corresponds to.*
*** The numbers under the 'PO Assessed' column represent the PO each CO corresponds to.*

PO1	1.1 Apply the knowledge of mathematics, science, engineering fundamentals to the solution of complex engineering problems.
	1.2 Apply the knowledge of an engineering specialization to the solution of complex engineering problems.
PO2	2.1 Identify, Research and Formulate complex engineering problems.
	2.2 Analyse and Reach substantiated conclusions using the principle of mathematics, the natural sciences and the engineering sciences

IX – Topics to be covered in Theory class*:

TOPICS	Specific Objective(s)	Time Frame	Suggested Activities	Teaching Strategy(s)	CO mapped
Mission & Vision of AIUB; Analyzing & Designing Algorithms,	Knowing Mission & Vision of AIUB. Mission & vision of AIUB, Formalize the students in thinking about designing and analyzing algorithms, gentle introduction to how we specify algorithms, some of the design strategies, many of the fundamental ideas used in algorithm	Week 1	Lecture and Lab: Review of different problems as basic data structure review, and their time complexities. Bubble Sort, Linear Search and Binary Search. **(Brainstorming on Vertex cover, Set cover and Independent set problem to get an**	Lecture, Notes/PPT Slides, Student Feedback, Board work, Exercise Solving, Question/ Answer Session.	CO1

	analysis (RAM model, Basic asymptotic notations, Lower bounds), Logarithmic-time, Linear-time, Polynomial- time, Exponential- time, NP-complete and Np-hard problems.		**idea on hardness of algorithms-only for discussion)**		
Searching & Sorting	Design & Analysis of different searching & sorting algorithms. Importance on space complexities.	Week 2	Lecture and Lab: Selection Sort, Insertion Sort, Counting Sort. ***QUIZ 1***	Lecture, Notes/PPT Slides, Student Feedback, Board work, Exercise Solving, Question/ Answer Session	CO1, CO2
Recurrences & Master Method	Forming Recurrence equations and how to solve them using repeated (backward) substitution method, substitution method, recursion-trees, master method. (Divide and conquer)	Week 3	Lecture and Lab: Restricting and Sorting Data Merge Sort, Quick Sort. (**Ackermann Function, Tower of Hanoi, Radix Sort and Finding Kth largest element from a sorted array- only for discussion.)** ***Lab Assessment 1***	Lecture, Notes/PPT Slides, Student Feedback, Board work, Exercise Solving, Question/ Answer Session.	CO1, CO2
Greedy strategy	Basic understanding about greedy strategy**, a**ctivity selection Problem, task scheduling problem, fractional knapsack problem, coin change, etc.	Week 4	Lecture and Lab: Knapsack (Fractional and 0/1) problems and other relevant Problems. ***QUIZ 2***	Lecture, Notes/PPT Slides, Student Feedback, Board work, Exercise Solving, Question/ Answer Session	CO1, CO2
Dynamic Programming	Definition and improvement of Dynamic Programming, Fibonacci problem, 0-1 Knapsack problem	Week 5	Lecture and Lab: Fibonacci Sequence and other Relevant Problems ***Lab Assessment 2***	Lecture, Notes/PPT Slides, Student Feedback, Board work, Exercise Solving, Question/	CO1, CO2

				Answer Session	
Discussion, Open problems and Brainstorming Session	Review, Discussion, Open problems and Brainstorming	Week 6	Session with open problems and Lab: ***Viva***	Lecture, Notes/PPT Slides, Student Feedback, Board work, Exercise Solving, Question/ Answer Session	CO3, CO4
Midterm Week Week 7					
Dynamic Programming	Matrix Chain Multiplication Problem, Longest Common Subsequence problems, etc.	Week 8	Lecture and Lab: Rod cutting, matrix chain multiplication, LCS.	Lecture, Notes/PPT Slides, Student Feedback, Board work, Exercise Solving, Question/ Answer Session	CO2, CO3, CO4
Graphs and Tress	Basics of graphs and tress along with their applications. Representation of Graphs, Basic search & traversal techniques, Depth-first Search, Breadth-first Search Topological Sort, Strongly Connected Component.	Week 9	Lecture and Lab: Link List, Adjacency Matrix, Adjacency List, BST, In order, Preorder and Post order traversal, Heap Sort. ***Assignment-1***	Lecture, Notes/PPT Slides, Student Feedback, Board work, Exercise Solving, Question/ Answer Session	CO2, CO3, CO4
Graphs Algorithms	Depth-first Search, Breadth-first Search Topological Sort, Strongly Connected Component.	Week 10	Lecture and Lab: BFS, DFS, Topological Sort. ***Lab Assessment Test-1 & QUIZ 1***	Lecture, Notes/PPT Slides, Student Feedback, Board work, Exercise Solving, Question/ Answer Session	CO2, CO3, CO4

Greedy Graph Algorithm	Minimum Spanning Tree, Prim-Jarnik Algorithm, Kruskal's Algorithm.	Week 11	Lecture and Lab: MST- Kruskal’s and Prim’s algorithms. ***Assignment 2***	Lecture, Notes/PPT Slides, Student Feedback, Board work, Exercise Solving, Question/ Answer Session	CO2, CO3, CO4
Shortest Path Algorithms	Single Source Shortest Path (Dijkstra’s Algorithm, Bellman-Ford Algorithm), All Pair Shortest Path (Floyd-Warshall Algorithm).	Week 12	Lecture and Lab: Shortest Path- Dijkstra’s and Warshall’s algorithms. ***Quiz 2***	Lecture, Notes/PPT Slides, Student Feedback, Board work, Exercise Solving, Question/ Answer Session	CO2, CO3, CO4
Discussion, Open problems and Brainstorming Session	Review, Discussion, Open problems and Brainstorming	Week 13	Session with open problems and Lab: Lab Exam ***Viva***	Lecture, Notes/PPT Slides, Student Feedback, Board work, Exercise Solving, Question/ Answer Session	CO3, CO4
Final term Week Week 14					

* The faculty reserves the right to change, amend, add or delete any of the contents.

XI- Course Requirements

At least **80% class attendance** is necessary to sit for the exam. If there is any assignment given to the students, they have to submit it before the deadline decided by the course teacher.

XII – Evaluation & Grading System

The following grading system will be followed but may vary on the components based upon OBE Rubrics.

Marking system for Theory Classes (Midterm and Final term)	
Quiz	20%
Attendance	15%
Assignment	15%
Performance	20%
Midterm/Final term exam	30%
Total	100%
Final Grade/ Grand Total	
Midterm:	40%
Final Term:	60%

Marking system for Laboratory Classes (Midterm and Final term)	
Lab Report	15%
Attendance	15%
Viva and Performance	20%
Midterm/Final term Test	50%
Total	100%
Final Grade/ Grand Total	
Midterm:	40%
Final Term:	60%

**** Since there is no assignment in Midterm, marks of the assignment component will be adjusted with Quiz and Performance components (5 marks with Quiz and 10 marks with Performance).**

XII ** – Evaluation & Grading System (Specially designed for online semester)

The following grading system will be followed but may vary on the components based upon OBE Rubrics.

Marking system for Theory Classes (Midterm and Final term)	
Quiz/Assessment 1 Quiz/Assessment 1	10% 10%
Attendance	10%
Assignment Performance	10% 20%
Midterm/Final term exam (MCQ) Midterm/Final term exam (Viva)	10% 30%
Total	100%
Final Grade/ Grand Total	
Midterm:	40%
Final Term:	60%

Marking system for Laboratory Classes (Midterm and Final term)	
Lab Report/Activity/Sincerity	15%
Attendance	15%
Viva and Performance	20%
Midterm/Final term Test	50%
Total	100%
Final Grade/ Grand Total	
Midterm:	40%
Final Term:	60%

Grand Total - - - - - - - - 40% of Midterm + 60% of Final Term

**** Midterm and final term ratio may change into 50%/50% based upon circumstances depending on satisfaction levels of the course teachers**

Letter	Grade Point	Numerical
90-100	A+	4.00
85 - < 90	A	3.75
80 - < 85	B+	3.50
75 - < 80	B	3.25
70 - < 75	C+	3.00
65 - < 70	C	2.75
60 - < 65	D+	2.50
50 - < 60	D	2.25
< 50	F	0.00

The evaluation system will be strictly followed as per the AIUB grading policy.

**** Course materials are designed and to be circulated among the respective teachers and the department duly.**

XIII – Teaching Methods

Maximum topics will be covered from the textbook. For the rest of the topics, reference books will be followed. Some Class notes will be uploaded on the web. White board will be used for most of the time.
For some cases, multimedia projector will be used for the convenience of the students.
Students must study up to the last lecture before coming to the class and it is suggested that they should go through the relevant chapter before coming to the class. Just being present in the class is not enough- students must participate in classroom discussions.

XIV – Textbook/ References

1) ***Introduction to Algorithms, Third Edition, Thomas H. Cormen, Charle E. Leiserson, Ronald L. Rivest, Clifford Stein (CLRS).***
2) *Fundamental of Computer Algorithms, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran (HSR)*
3) *Helpful link for Problem Solving :* ***http://acm.uva.es/problemset/***
4) *Lectures and Laboratory works will be provided online at the course website weekly.*

XV - List of Faculties Teaching the Course

1) Md. Manzurul Hasan, Assistant Professor
2) Dr. Ashraf Uddin, Assistant Professor
3) Dr. Md. Asraf Ali, Faculty

XVI – Verification:

Prepared by --------------------------------- Md. Manzurul Hasan Course Convener Date-	Checked and certified by: .... (Asst. Prof. Dr. Md. Mahbub Chowdhury Mishu) (Head (In Charge) of Department for UG program, CS) Date:...	Approved by: .... Asso. Prof. Mashiour Rahman (Asso. Dean of Faculty of FST) Date:..
	Moderated by : …………………….. Date : ………………………..	Moderated by : ………………………. Date : ………………………..

APPENDIX

Program Outcomes (POs)

PO1	Engineering Knowledge
1.1	Apply the knowledge of mathematics, science, engineering fundamentals to the solution of complex engineering problems.
1.2	Apply the knowledge of an engineering specialization to the solution of complex engineering problems
PO2	**Problem Analysis**
2.1	Identify, Research and Formulate complex engineering problems
2.2	Analyze and Reach substantiated conclusions using the principle of mathematics, the natural sciences and the engineering sciences

Mapping of CO Assessment Method and Rubric

The mapping between Course Outcome(s) (COs) and The Selected Assessment method(s) and the mapping between Assessment method(s) and Evaluation Rubric(s) is shown below:

CO	Description	Learning Domain	Assessment Method	Assessment Rubric
CO1	Outline terminologies and methods of classical algorithms analysis and design	Cognitive	Quiz	Rubric for Quiz-2

	to solve some conventional/real life algorithmic problems.			
CO2	Demonstrate the selective well known and self-developed (if any) algorithms in efficient ways.to solve some engineering problems.	Cognitive	Midterm/Final	Rubric for Midterm
CO3	Break down some complex realistic problems into modular problems and their solutions.	Affective	Assignment	Rubric for Assignment 1
CO4	Justify the usage of the algorithms and their data structures to solve some complex realistic problems along with complexity analysis.	Affective	Assignment	Rubric for Assignment 2

Rubric for Quiz 2 Assessment (CO1)

Marking Criteria	**Marks Distribution (Maximum 5 x 3=15)**				**Acquired Marks**
	Inadequate (0-.9)	**Satisfactory (1-1.9)**	**Good (2-2.9)**	**Excellent (3)**	
Basic Understanding	• Students are not familiar with the terms.	• Students can understand the definitions	• Students can understand the definitions with relevant examples	• Concept is clear with proper use of them.	
Depth on Discrete Math & Data Structure	• Basic knowledge is not clear as fundamental prerequisites.	• Depth is clear on the fundamental contents.	• Students can apply them.	• Students can apply whatever problem is assigned.	
Problem Solving Skill	• Strength is weak.	• Strength is satisfactory but only on the fundamental problems.	• Strength is good with nontrivial problems.	• Students can brainstorm on any problem whatever is assigned and can come to a near conclusion.	
Understanding fundamental and essential algorithms on different subject matters	• Not at all or very superficial.	• Nearly clear.	• Clear with enough examples.	• Students can reduce the problems into real life problems.	
Implementation Skill	• Not at all or	• Very basic.	• Can	• Can	

	very superficial.	Students can build the skeleton	implement.	implement and are able to change the data structure.	
				Acquired Marks:	
				CO Pass / Fail:	

Rubric for Midterm/Final Assessment (CO2)

Marking Criteria	**Marks Distribution (Maximum 5 x 3=15)**				**Acquired Marks**
	Inadequate (0-.9)	**Satisfactory (1-1.9)**	**Good (2-2.9)**	**Excellent (3)**	
Basic Understanding	• Students are not familiar with the terms.	• Students can understand the definitions	• Students can understand the definitions with relevant examples	• Concept is clear with proper use of them.	
Depth on Discrete Math & Data Structure	• Basic knowledge is not clear as fundamental prerequisites.	• Depth is clear on the fundamental contents.	• Students can apply them.	• Students can apply whatever problem is assigned.	
Problem Solving Skill	• Strength is weak.	• Strength is satisfactory but only on the fundamental problems.	• Strength is good with nontrivial problems.	• Students can brainstorm on any problem whatever is assigned and can come to a near conclusion.	
Understanding fundamental and essential algorithms on different subject matters	• Not at all or very superficial.	• Nearly clear.	• Clear with enough examples.	• Students can reduce the problems into real life problems.	
Implementation Skill	• Not at all or very superficial.	• Very basic. Students can build the skeleton	• Can implement.	• Can implement and are able to change the data structure.	
				Acquired Marks:	

CO Pass / Fail:	

Rubric for Assignment 1 Assessment (CO3)

Marking Criteria	**Marks Distribution (Maximum 8 x 2=16 to be converted into 15)**				**Acquired Marks**
	Inadequate (0-2)	**Satisfactory (3)**	**Good (4)**	**Excellent (5)**	
Analysis Skill	• Irrelevant	• Relevant	• Relevant with good design	• Good design with appropriate roadmap	
Problems mapping skill	• Vague reduction.	• Mapping but poor analysis	• Sound mapping but does not cover all parameters	• Mapping meeting all parameters.	
Modular Design	• Does not suit.	• Modules are not accurate.	• Accurate but not corelated	• Accurate with corelated	
Subcomponents solutions	• Solutions are not correct.	• Nearly solved.	• Solved but not appropriate.	• Solved with justification.	
Solutions Merging	• Merge fail	• Merge done but not consistent.	• Consistent but not case analyzed.	• In all cases gives accurate solution.	
Efficiency comparisons	• Efficiency is not done.	• Efficiency done but no comparison	• Comparison done but not better.	• Comparison done with better improvement.	
Justification	• Not justified.	• Poorly justified.	• Justified but does not fulfill all criteria.	• In all criteria it is justified	
Discovery of Knowledge	• No discovery	• Discovered but poor.	• Moderately discovered.	• Novel discovery.	
				Acquired Marks:	
				CO Pass / Fail:	

Rubric for Assignment 2 Assessment (CO4)

Marking Criteria	**Marks Distribution (Maximum 8 x 2=16 to be converted into 15)**				**Acquired Marks**
	Inadequate (0-2)	**Satisfactory (3)**	**Good (4)**	**Excellent (5)**	
Analysis Skill	• Irrelevant	• Relevant	• Relevant with good design	• Good design with appropriate roadmap	
Problems mapping skill	• Vague reduction.	• Mapping but poor analysis	• Sound mapping but	• Mapping meeting all	

			does not cover all parameters	parameters.	
Modular Design	• Does not suit.	• Modules are not accurate.	• Accurate but not corelated	• Accurate with corelated	
Subcomponents solutions	• Solutions are not correct.	• Nearly solved.	• Solved but not appropriate.	• Solved with justification.	
Solutions Merging	• Merge fail	• Merge done but not consistent.	• Consistent but not case analyzed.	• In all cases gives accurate solution.	
Efficiency comparisons	• Efficiency is not done.	• Efficiency done but no comparison	• Comparison done but not better.	• Comparison done with better improvement.	
Justification	• Not justified.	• Poorly justified.	• Justified but does not fulfill all criteria.	• In all criteria it is justified	
Discovery of Knowledge	• No discovery	• Discovered but poor.	• Moderately discovered.	• Novel discovery.	
Acquired Marks:					
CO Pass / Fail:					