On completion of the assignment you should be able to demonstrate the following learning outcomes:

CLO2	Develop a solution for a problem implementing appropriate data structure. (C4, PLO2)	Project - Artefact
CLO3	Clearly present the solution developed (A4, PLO5)	Project - Documentation & Presentation

Test Specification Table :

Group Project : Artefact (40%)

Question No.	Topic	Question Vs Taxonomy						PLO
		Cognitive Level						
		1	2	3	4	5	6	
		SQ	SQ	SQ	SQ	SQ	SQ	
1	Design				25%			2
2	Implementation				75%			2
	Total				**100%**			

Group Project : Documentation and Presentation (10%)

Question No.	Topic	Question Vs Taxonomy						PLO
		Affective Level						
		1	2	3	4	5		
		SQ	SQ	SQ	SQ	SQ		
1	Documentation				50%			5
2	Presentation				50%			5
	Total				**100%**			

PERFORMANCE CRITERIA 1

CLO / PLO	ASSESSMENT CRITERIA		FAIL	MARGINAL FAIL	PASS	CREDIT	DISTINCTION
CLO 2 - PLO2	Design (20%)	Data Structure Proposal (10 marks)	**0 – 3**	**4**	**5 - 6**	**7**	**8 - 10**
			Illogical data structure proposed with poor justification.	Logical data structure proposed with poor justification.	Logical data structure chosen with fair justification.	Logical data structure chosen with good justification.	Logical data structure chosen with excellent justification.
		Algorithms - sort and search **(15 marks)**	**0 – 5**	**6 - 7**	**8 - 9**	**10 - 11**	**12 - 15**
			Illogical sort and search algorithms proposed with acceptable algorithms presentation. Poor algorithm descriptions.	Logical sort and search algorithms proposed with poor presentation. Poor algorithm description.	Logical sort and search algorithms proposed with major errors in algorithm presentation. Fair algorithm descriptions with missing efficiency discussion and minimum comparative analysis.	Logical sort and search algorithms proposed with minor errors in algorithm presentation. Good algorithm descriptions with proper efficiency discussion and some comparative analysis.	Logical sort and search algorithms proposed with correct presentation of algorithm. Excellent algorithm descriptions with proper efficiency discussion and good comparative analysis.
	Implementation (60%)	Program (65 marks)	**0 - 25**	**26 - 35**	**33 - 42**	**43 - 48**	**49 - 65**
			Program is not given Program does not compile or run. Less than 30% of the required functionalities are implemented. Implementation uses very inefficient data structures or algorithms.	Program does not compile or run, but coding logic is almost correct. Between 30% and 50% of the required functionalities are correctly implemented. Implementation uses inefficient data structures or algorithms.	Program compiles perfectly and executes. Between 50% and 70% of the required functionalities are correctly implemented. Implementation uses inefficient data structures or algorithms which is	Program compiles and executes. Between 70% and 90% of the required functionalities are correctly implemented. Implementation uses data structure(s) or algorithm that is not most efficient and slightly differ from the proposed design.	Program compiles and executes perfectly. At least 90% of the required functionalities are correctly implemented. Efficient data structure(s) and\or algorithms are used in the implementation according to the proposed design.

			No proper code structure and no comments provided.	Unclear coding style, and no comments provided. Functionalities are not tested/validated in program execution.	different from the proposed design. Unclear coding style, or code is not properly commented. Functionalities are not full tested/validated in program execution, or produce errors in some cases.	Clear coding style, and code is properly commented. Functionalities are not fully tested/validated in program execution.	Clear coding style and structure, and code is properly commented. Functionalities are fully tested/validated in program execution.
		Adherence (10 marks)	**0 – 3**	**4**	**5 - 6**	**7**	**8 - 10**
			Completely different implementation from the proposed data structures and algorithms.	Poor implementation of the proposed data structures. Only 1 – 2 of the proposed algorithms implemented.	Fair implementation of the proposed data structures. Missing 1 – 2 implementations of the proposed algorithms.	Good implementation following the proposed data structures. All proposed algorithms are properly implemented.	Excellent implementation of the proposed data structures with proper revision. All proposed algorithms are properly implemented.

PERFORMANCE CRITERIA 2

CLO / PLO	ASSESSMENT CRITERIA		FAIL	MARGINAL FAIL	PASS	CREDIT	DISTINCTION
CLO 3 - PLO 5	**Documentation and Presentation (20%)**	**Documentation (50 marks)**	**0 - 19**	**20 - 24**	**25 - 32**	**33 - 37**	**38 - 50**
			No documentation submitted. Documentation merely contains the cover page and printout of the source code.	Incomplete documentation with missing major component. Poor layout / flow. No referencing (if applicable).	Document missing minor components. Average layout / flow. Incomplete referencing. Incomplete input/output sample with improper explanation/ description.	Good layout/flow. No missing components in the documentation. Good documentation standards. Adhered to the referencing standards with minor errors.	Excellent layout / flow No missing components of the documentation Excellent documentation standards.

						Incomplete input/output samples with proper explanation / description.	Adhered to the referencing standards. Complete input/output sample with clear explanation/ description
		Presentation (50 marks)	**0 - 19**	**20 - 24**	**25 - 32**	**33 - 37**	**38 - 50**
			Did not turn up for presentation Unable to trace any of the code / work done Unable to execute the program Unable or barely able to answer any of the question asked	Barely able to trace the code / work done Having difficulties in running the codes Mostly inaccurate / illogical answer / explanation provided Barely able to answer questions asked	Able to trace some code / work done Able to execute the program properly Able to answer some questions posed accurately / logically	Able to trace the code / work done Able to execute the program properly Able to explain and show a good understanding of how the program works Able to answer most questions posed accurately	In depth understanding of the code / work done Able to execute the program Able to explain and argue the work submitted Show additional concepts / new ideas used in the solution Able to answer all questions posed with minimal omissions