

CSE 3223: Information System Design and Software Engineering

Part A

1. Course No./Course Code: **CSE 3223 (BNQF 0613)**
2. Course Title: **Information System Design and Software Engineering**
3. Course Type (GEEd/Core Course/Elective): **Core Course**
4. Year/Level/Semester/Term: **Year 3, Semester 2**
5. Academic Session: **N/A**
6. Course Teacher/Instructor: **N/A**
7. Prerequisite(s) (if any): **CSE 3103**
8. Credit Value: **3**
9. Credit Hours: **3**
10. Total Marks: **100**
11. Rationale of the Course: **SDG 4 (Ensure Technical Level Education) and SDG 8 (Reduce Unemployment Rate).**
12. Faculty: **Engineering**
13. Department: **Computer Science and Engineering (CSE)**
14. Programme: **Bachelor of Science in Computer Science and Engineering (B.Sc. in CSE)**
15. Course Objectives:
Information System Design:
Information and System; Systems Analysis and Systems Analyst; Information gathering techniques; Structured analysis of systems; Feasibility Study: Concepts (abstraction, refinement, modularity and hierarchy) and classification, Introduction to modeling language (Use case diagram, Sequence diagram and Activity diagram), Cost benefit analysis; Project scheduling; System design techniques; User interface design.
Software Engineering:
Introduction to system engineering and software engineering; Software requirements analysis, modeling and specification; Software Designing: principles, models, design patterns and specification; Software testing: objectives and principles, testability, testing design and implementation models and documentations, verification, validation and debugging; Quality factors and metrics for different software engineering phases; Software project management issues.
16. Mapping of Course Outcomes with Bloom's Taxonomy and Programme Outcomes
After the successful completion of this course, students are expected to be able to:

Sl. No.	Cos	POs	Bloom's Taxonomy		
			C	A	P
1	Explain different development principles for management necessities	4	2		
2	Implement efficient practices for ensuring improved application systems	12	3		
3	Differentiate between available alternatives for efficient decision making	2	4		

17. Mapping of COs with Knowledge Profiles, Complex Engineering Problem Solving and Complex Engineering Activities

Course Outcome	Knowledge Profile	Complex Problem Solving	Complex Engineering Activities
CO1	K4		
CO2	K6		
CO3	K6		

Part B

18. Week-wise Course Plan

Week	Topics	Teaching-Learning Strategy	Assessment Strategy	Corresponding COs
1	Overview of the course; Introduction to System Analysis, Systems Analyst's Role and Development methodologies. Project Management (Objective, Project initiation, Problems in the project organization, Selection of Project, Feasibility Analysis, Needs & Requirement, Cost & Benefit).	- Lecture - Brainstorming Session	- <i>Class Performance</i>	1

Week	Topics	Teaching-Learning Strategy	Assessment Strategy	Corresponding COs
2	Comparing Cost & Benefits, Cash flow Analysis, Present Value Analysis, Gantt Chart. PERT (Program Evaluation & Review Technique) Diagram, Critical Path Analysis, using Gantt Chart for Project Scheduling / Using PERT Diagrams.	- Lecture - Brainstorming	- <i>Class Performance</i>	1,2,3
3	Information Gathering Interactive Methods (Steps in Interview Preparation, Question Types, Joint Application Design with its benefits and drawback) Different Questions Pattern, Using Scales in Questions, Designing Questions	- Lecture - Think-Pair-Share (TPS)	- <i>Class Performance</i> - <i>Quiz 1</i>	1,3
4	Information Gathering: Unobtrusive Methods (Sampling, Needs of Sampling, Sampling Design) Information Gathering: Unobtrusive Methods (Sampling Size, Decision, Analyzing Quantitative Documents, Observing Behaviors	- Lecture - Brainstorming Session	- <i>Class Performance</i>	1,3
5	Data Flow Diagram (DFD): Notations, External Entity, Process, Data Store, Data Flow, How to draw a DFD from requirements analysis. Context Level Diagram, Level 0 Diagram, Level 1 Diagram. Object-Oriented System Analysis and Design	- Lecture - Brainstorming	- <i>Class Performance</i>	2
6	Using UML Object-Oriented Analysis, Goals of Object Oriented Analysis, Object Oriented Concepts, Class Symbols, Inheritance, CRC Cards, Creating, CRC Cards, UML Concepts, Use Case Modeling, Activity Diagrams, Sequence & Communication Diagrams, Importance of Using UML for Modeling Object Modeling, Relationship among Classes,	- Lecture - Think-Pair-Share (TPS)	- <i>Class Performance</i>	2

Week	Topics	Teaching-Learning Strategy	Assessment Strategy	Corresponding COs
	Class Diagrams, and Review of System Analysis.			
7	Testing Conventional Applications: alpha testing, beta testing, system testing, debugging, Testing Conventional Applications: white box testing, black box testing, Cyclomatic complexity analysis for finding independent path for testing, gray box testing.	- Lecture - Brainstorming Session	- <i>Class Performance</i> - <i>Quiz 2</i>	2
Mid Break				
8	Software and Software Engineering Products, software costs, wear out vs deteriorate, software applications, layered technology, umbrella activities, Software Process Model (Development Models: Waterfall Model, Incremental Process Model, Rad Model, Evolutionary Process Models).	- Lecture - Think-Pair-Share (TPS)	- <i>Class Performance</i>	1,2,3
9	Software Process Model (Prototyping Model, Spiral SDLC, Model, Concurrent Development Model), The Unified Process Model (UP), Different Phases of UP, Agile Development (An Agile View of Process, Agile Process Models, Extreme Programming-XP).	- Lecture - Think-Pair-Share (TPS)	- <i>Class Performance</i>	1,2,3
10	Estimation for Software Projects: project planning, task for project planning, software scope, resource estimation, categories of resources	- Lecture - Think-Pair-Share (TPS)	- <i>Class Performance</i>	1,2,3
11	Different Estimation Techniques for Software Projects, LOC based software estimation, FP count based software estimation and solving math.	- Lecture - Brainstorming Session	- <i>Class Performance</i> - <i>Quiz 3</i>	1,2,3

Week	Topics	Teaching-Learning Strategy	Assessment Strategy	Corresponding COs
12	Agile Process Models-Scrum Framework, Crystal Methodology, Software Testing Strategies: Strategy, validation and verification, unit testing, integration testing, validation, system testing, ensuring successful testing, Driver, Stub, Software Testing Strategies: Different types of Integration testing, sandwich testing, regression testing, smoke testing, testing strategies for object oriented testing, validation testing.	- Lecture - Think-Pair-Share (TPS)	- <i>Class Performance</i>	2, 3
13	Different Software Design Patterns, Risk Management, Review on Agile process model for software development	- Lecture - Brainstorming Session	- <i>Class Performance</i>	2,3
14	Review Class	- Lecture	- <i>Class Performance</i> - <i>Assignment</i>	1, 2,3

Part C

19. Assessment and Evaluation

- 1) Assessment Strategy: ***Class Performance, Quizzes/Assignments, and Final Examination***
- 2) Marks distribution:
 - a) Continuous Assessment: ***Class Performance (10), Quizzes/Assignments (20)***
 - b) Summative: ***Final Examination (70)***
- 3) Make-up Procedures: ***Carryover/Clearance/Improvement Examination***

Part D

20. Learning Materials

20.1. Required (if any)

1. Software Engineering (5th Edition). Authored by: Roger S. Pressman
Publisher: McGraw-Hill Higher Education.

2. System Analysis and Design (8th Edition). Authored by: Kenneth E. Kendall And Julie E. Kendall.

Publisher: Pearson Education.

20.2. Recommended (if any)

20.3. Others (if any)