Course Title: Object Oriented Analysis and Design **Credit: 3**
Course No: CSIT.413 **Number of period per week: 3+3**
Nature of the Course: Theory + Lab **Total hours: 45+45**
Year: Fourth, Semester: Seventh
Level: B. Sc. CSIT

1. Course Introduction

This course starts with object oriented concepts and moves towards the preparation of standard UML diagrams using an UML modeling tool. Besides this the course covers details of object oriented analysis and design process.

2. Objectives

By the end of this course, students will be able

- → To learn the concept of Object Oriented Software Development Process
- → To get acquainted with UML Diagrams
- → To understand Object Oriented Analysis Processes
- → To understand Object Oriented Design Processes

3. Specific Objectives and Contents

Specific Objectives	Contents
• Discuss importance of object orientation • Exemplify objects, classes, inheritance, polymorphism, and aggregation • Explain object attributes, object state, and object identity • Understand object oriented system development process	**Unit I: Overview of Object Oriented System (8 hr Hrs)** 1.1. Orthogonal views of software, Why object orientation, Overview of Unified approach 1.2. Object, Class, Object attribute and state, Object methods and behaviour, Messages 1.3. Encapsulation and Information Hiding, Class Hierarchy, Polymorphism, Object relationships-association, aggregation and composition 1.4. Object Identity, Static and Dynamic Binding, Object Persistence, Meta Classes 1.5. Object Oriented System Development: Object Oriented Analysis, Object Oriented Design, Prototyping, Implementation, Incremental Testing
• Discuss different modeling techniques, methodologies and identify their strengths and drawback • Understand concepts and importance of patterns and	**Unit II: Methodologies, Modeling and UML (10 hr)** 2.1. Object Oriented Methodologies: Rumbaugh Modelling Techniques, Booch methodology, Jacobson Methodologies 2.2. Patterns and its Types, Anti-patterns, Pattern Templates, Frameworks 2.3. UML: Static and dynamic models, Introduction of

framework • Draw different UML diagrams to model some system	UML, Importance of Modelling 2.4. UML Diagrams: Class Diagram, Object Diagram, Use-case Diagram, Interaction Diagrams, Sate-chart diagrams, activity diagram, implementation Diagrams
• Describe object oriented analysis and its difficulties • Understand object oriented analysis process • Perform OOA some real world system to identify actors, use cases, classes, methods and attributes	**Unit III: Object Oriented Analysis (10 hr)** 3.1. Introduction, Analysis Difficulties, OOA Process, Finding actors, Finding Use cases, Naming Use cases, uses and extends association, Case Study for finding use cases and actors 3.2. Object Analysis: Classification theory, Approaches for finding classes: Noun phrase approach, Common class pattern approach, Use case driven approach, CRC approach 3.3. Identifying Relationships: Associations, Super-Sub Class Relationships, Aggregation, Identifying Attributes and Methods, Case Study on Identifying Relationships, Methods and Attributes
• Differentiate OOD from OOA • Understand OOD process, Axioms, and corollaries • Discuss and exemplify object relational mapping • Explain access layer and view layer design process • Perform OOD of some real world system	**Unit IV: Object Oriented Design (12 hr)** 4.1. OOD Process, OOD Axioms and Corollaries, Design patterns, UML Object Constraint Languages, Designing Classes, Define Visibility, Refine Attributes, Designing Methods and Protocols 4.2. Access Layer, Object Relational Databases, Object Relational Mapping, Process for designing access layer classes 4.3. View Layer, Process of designing view layer classes, Macro level process, Micro level process, UI design rules 4.4. Case Study on designing business layer, access layer and view layer classes.
• Discuss and exemplify different types of errors • Explain different test strategies and understand impact of reusability in testing • Exemplify usability testing and test cases.	**Unit V: Software Quality (5 Hrs)** 5.1. Quality Assurance Tests, Testing Strategies, Impact of Object Orientation on Testing, Test Cases, Test Plan, Continuous Testing 5.2. Verification and Validation, Usability Testing, case study on Usability Test Plan and Test Cases

Evaluation System

Undergraduate Programs

External Evaluation	Marks	Internal Evaluation	Weightage	Marks	Practical	Weightage	Mark
End semester examination	60	Assignments	20%	20	Practical Report copy	25%	20
(Details are given in the separate table at the end)		Quizzes	10%		Viva	25%	
		Attendance	20%		Practical Exam	50%	
		Internal Exams	50%				
Total External	60	Total Internal	100%	20		100%	20
Full Marks 60+20+20 = 100							

External evaluation

1. **End semester examination:**
   It is a written examination at the end of the semester. The questions will be asked covering all the units of the course. The question model, full marks, time and others will be as per the following grid.

2. **External Practical Evaluation:**
   After completing the end semester theoretical examination, practical examination will be held. External examiner will conduct the practical examination according to the above mentioned evaluation. There will be an internal examiner to assist the external examiner. Three hours time will be given for the practical examination. In this examination Students must demonstrate the knowledge of the subject matter.

Full Marks: 100, Pass Marks: 45, Time: 3 Hrs

Nature of question	Total questions to be asked	Total questions to be answered	Total marks	Weightage
Group A: multiple choice*	20	20	20×1 = 20	60%
Group B: Short answer type questions	7	6	6×8 = 48	60%
Group C: Long answer type questions	3	2	2×16 =32	60%
			100	100%

Each student must secure at least 50% marks in internal evaluation in order to appear in the end semester examination. Failed student will not be eligible to appear in the end semester examinations.

Internal evaluation

Assignment: Each student must submit the assignment individually. The stipulated time for submission of the assignment will be seriously taken.

Quizzes: Unannounced and announced quizzes/tests will be taken by the respective subject teachers. Such quizzes/tests will be conducted twice per semester. The students will be evaluated accordingly.

Attendance in class: Students should regularly attend and participate in class discussion. Eighty percent class attendance is mandatory for the students to enable them to appear in the end semester examination. Below 80% attendance in the class will signify NOT QUALIFIED (NQ) to attend the end semester examination.

Presentation: Students will be divided into groups and each group will be provided with a topic for presentation. It will be evaluated individually as well as group-wise. Individual students have to make presentations on the given topics.

Mid-term examination: It is a written examination and the questions will be asked covering all the topics in the session of the course.

Discussion and participation: Students will be evaluated on the basis of their active participation in the classroom discussions.

Instructional Techniques: All topics are discussed with emphasis on real-world application. List of instructional techniques is as follows:

- Lecture and Discussion
- Group work and Individual work
- Assignments
- Presentation by Students
- Quizzes
- Guest Lecture

Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class (es), it is his/her sole responsibility to cover the topic(s) taught during that period. If a student fails to attend a formal exam/quiz/test, there won't be any provision for re-exam. Unless and until the student clears one semester he/she will not be allowed to study in the following semesters.

Laboratory Work

Student should use some case tools to draw UML diagrams discussed the course. Besides this student should perform analysis and design of some small scale some real world system and needs to make report.

Prescribed Text

- Ali Bahrami, Object Oriented Systems Development using the Unified Modeling Language, McGraw Hill, Reprint 2009.

References

- Grady Booch, James Rumbaugh, Ivar Jacobson, "*The Unified Modeling Language User Guide*", 2nd Edition, Pearson Education, 2007.
- Bernd Oestereich, Developing Software with UML, Object-Oriented Analysis and Design in Practice, Addison-Wesley, 2000.