

## COURSE INTRODUCTION

# OBJECT ORIENTED SOFTWARE ENGINEERING

(OOSE Fall 2015)

## EXECUTIVE SUMMARY

Designing and developing software systems for complex workflow automation, decision-making and collaboration, this course is designed to prepare post-graduate students to use object oriented approach for software requirement analysis, design, development, and testing for quality software development.

## COURSE DESCRIPTION

The objective of the course is to introduce students to object oriented software engineering practices and master necessary skills to produce good quality software. The course starts with foundation concepts on software engineering then progresses through various OOSE activities. Students develop skills on UML, Rational Unified Process, Open Unified Process, CASE tools, and software quality assurance.

Upon completion of this course, students will have a good understanding of

- object orientation and object oriented thinking,
- how object oriented analysis can be used to decompose and understand complex systems,
- object oriented design, implementation and testing methods,
- software project planning and risk management, and
- maintaining software quality.

## COURSE OUTLINE:

**COURSE TITLE** : **Object Oriented Software Engineering** (Last Updated on Saturday, January 14 2006)  
**COURSE CODE** : COM615.3  
**Credit** : 3  
**Class Load** : 3 hours/week  
**Evaluation** :

	<b>Theory</b>	<b>Classroom participation</b>	<b>Class tests</b>	<b>Presentation</b>	<b>Project</b>	<b>Total</b>
<b>Sessional</b>	20	10	5	10	15	60
<b>Final</b>	40					40
<b>Total</b>	70	10	5	10	15	100

The course objective is to provide required knowledge on the various issues of software project management and related tasks including planning, design, development, implementation, maintenance and cross life cycle activities using object oriented concepts and models.

## **COURSE CONTENTS:**

- 1. Introduction (3 hrs)**  
History of software engineering, SDLC, Various Software Process Models (Linear, prototyping, RAD, Evolutionary, Waterfall, Incremental, Spiral, etc), Complex System Structure and Design
- 2. Project Management Planning & Risk Analysis (5 hrs)**  
Four P's (People, Product, Process, & Project), Software scope, Feasibility: Importance, Feasibility assessment, Economic, Technical, Operational, and Schedule Feasibility, Resources: human, reusable software, environment, Project Estimation, The make/buy decision, outsourcing, Project scheduling tracking, Risk Management Strategies, Software Risks, Risk Identification, Projection and Assessment
- 3. Object Orientation Fundamentals and UML (8 hrs)**  
Introduction to OO Analysis and Design, Software Industries Best Practices, Defining models, Use Cases: Object Oriented Development Cycle, Overview of the Unified Modeling Language: UML fundamentals and Notations, UML for Real Time Systems (UML RT), Real time System Artifacts, Unified Process
- 4. Object Oriented Analysis (6 hrs)**  
Requirements Analysis, Requirements Gathering and Modeling: FURPS+ and JAD, Interview techniques and Questionnaires, Building Conceptual model, Adding associations and Attributes, Representation of System Behavior, Use case Details, System Sequence Diagrams, Operations Contracts
- 5. Object Oriented Design (6 hrs)**  
Analysis to Design , Describing and Elaborating Use Cases Interaction Diagram, Objects and Design Patterns, Design Patterns for Real-time Systems, Determining Visibility, Design Class Diagram
- 6. Object Oriented Implementation (3 hrs)**  
Programming and Development Process, Mapping Design to Code, Creating Class Definitions from Design Class Diagrams. Creating Methods from Collaboration Diagram, Updating Class Definitions, Classes in Code, Exception and Error Handling
- 7. Software Quality Issues (3 hrs)**  
Classification of software qualities, representative qualities, quality requirements based on application areas, Quality Concepts, Software Quality Assurance, Software Reviews, Formal Technical Reviews, Software Reliability, Software Configuration Management, SCM Standards.
- 8. Software Testing Techniques (3 hrs)**  
Testing Fundamentals, Test Case Design, White Box Testing, Basis Path Testing, Control Structure Testing, Black-Box Testing, Unit Testing, Integration Testing, Validation Testing, System Testing
- 9. Software Quality Assurance and Configuration Management (5 hrs)**  
Concepts, Software Quality Assurance, Software Reviews, Formal Technical Reviews, Formal Approaches to SQA, Statistical Quality Assurance, Software Reliability, ISO 9000 Quality Standards, SQA Plan, Software Configuration Management, SCM Process,

Identification of Objects in the Software Configuration, Version Control, Change Control, Configuration Audit, Status Reporting, SCM Standards

**10. Case Study:**

**(2 hrs)**

An individual case study should be given to each student on software project and should be analyzed with any UML CASE tool (like *Rational Rose Enterprise Suit 2000*) and implemented in OO. 25 % to 50% of sessional marks should be allocated for evaluation.

**REFERENCES:**

1. R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 5/e. McGraw Hill International Edition
2. G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language - User Guide*, Addison-Wesley
3. C. Ghazi, M. Jazayeri, and D. Mandrioli, *Fundamentals of software Engineering*, Prentice Hall of India, Ltd.
4. Grady Booch, *Object Oriented Analysis and Design with Applications*, Second Edition, Pearson
5. C. Larman, *Applying UML and Patterns*, Pearson
6. R. Fairly, *Software Engineering*, McGraw Hill Publishing Co.
7. Other additional references will be provided as per need.

**SUPPLEMENTS AND REFERRED MATERIALS:**

Available at <http://www.pramodparajuli.com/teaching/oose-ncit>. Eclipse with UMLet plugin will be used as CASE tool.

**LECTURES:**

Classroom meetings for lectures will be held for once every week. Each meeting will last for 3 hours with short break in between. These meetings involve lectures on course contents, discussions, practices, presentations etc.

**STUDENT'S PARTICIPATION:**

- Students need to give full participation in the classroom activities.
- Students will collaborate with peers for various group activities/assignments.
- Fluent oral and written communication skills are required for classroom interactions, assignments, and examination.
- Different activities such as group discussions and case studies will be conducted at the end of every chapter.
- Students will undertake project work to develop objected oriented solution for innovative software projects. A comprehensive report should be developed and formal presentation should be made for evaluation purposes.