CMP 388.3 Object Oriented Software Engineering (3-1-0)
Amended Course as passed by the Subject Committee Meeting held on Feb. 29, 2004.

	Theory	Practical	Total
Sessional	50		50
Final	50		50
Total	100		100

Course Objectives:

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** **(5hrs)**

History of software engineering, role of software engineering in system design. The software development life cycle . Relationship of software engineering to other areas of computer science and other disciplines . Software Process Models, Linear Model. The Prototyping Model, The RAD Model, Evolutionary Models, Waterfall Model, Incremental Model, Spiral Model etc.

2. **Project Management Concepts** **(2hrs)**

The Management Spectrum, People, The Product, The Process, The project.

3. **Project Planning** **(5hrs)**

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

4. **Risk Analysis and Management** **(3hrs)**

Software risks, Risk Identification, Risk Projection, Risk Refinement, risk Mitigation, Monitoring and Management, Safety Risks and Hazards.

5. **Software Qualities** **(4hrs)**

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.

6. **System Engineering Principles** **(5hrs)**

Business Process Engineering, Product Engineering, Requirements Engineering, Requirements Elicitation, Analysis and Negotiation, Specification, Validation, and Management, System Modeling (in detail): Data (ERD), Functional (DFD, CFD), Behavioral (STD) etc.

7. **Software Testing** **(4hrs)**

Software Testing fundamentals, Importance of Testing, Test Case Design White Box Testing, Black Box Testing, Unit Testing, Integration Testing, Validation Testing, System Testing Verifying and Debugging, Symbolic execution.

8. **Object Oriented Fundamentals** **(2hrs)**

OO Paradigm: Classes, objects, attributes, operations, methods, services, messages, encapsulation, inheritance, polymorphism, etc.

9. **UML** **(6hrs)**

Unified Modeling Language Fundamentals, UML, notations, Structural Models: (concepts: class, object, relationship, interfaces, packages, instances) Class Diagram, and Object Diagram, Behavioral Models (concepts: interactions, scenarios, use cases, event, signal, process), Use Case Diagram, Interaction Diagram, Collaboration Diagram, State Transition Diagram, Activity Diagram, Architectural Models (concepts: components, deployment, collaboration, patterns, etc), Component Diagram, Deployment Diagram, UML, based any CASE tool and its use.

10. **Object Oriented Analysis** **(5hrs)**

OO based Analysis, Unified approach, Domain and reuse analysis, Identification of class and objects, Identification of class and object semantics, Use case and CRC modeling, Structure, hierarchy, subject and subsystems definition, Identify class and object relationship, Object- relationship modeling Events and states identification, State representation, Systems behavior representation

11. **Object Oriented Design** **(4hrs)**

Design Issues, Unified Approach to design, Partitioning of analysis model, Concurrency and subsystem allocation, task management component. User interface component, Data management component, Resource management component, Inter-subsystem, Communication, Object description, Data structure, Component and interfaces, Design Patterns and reuse, Elaboration and implementation of Use cases Class, Object collaboration, Interaction, STD diagram etc.

Case Study :

An individual case study should be given to each student on software project and should be analyzed with UML CASE tool and implemented in OO.10% of sessional marks should be allocated for evaluation.

Reference Books

1. R.S Pressman, Software Engineering: A Practitioner's Approach, 5/e, Mc Graw Hill International Edition
2. G. Booch, J.Rumbaugh, J. Jacobson, The Unified Modeling Language –User Guide Addison – Wesley
3. C.Ghezzi, M. Jazayeri and D. Mandrioli, Fundaments of Software Engineering prentice Hall of India, Ltd.
4. G. Booch, Object Oriented Analysis and Design with Applications 2/e Pearson
5. C. Larman, Applying UML and patterns, Pearson
6. R. Fairly, Software Engineering, Mc Graw Hill Publishing Co.