

BIT 420: OBJECT ORIENTED SYSTEMS ANALYSIS AND DESIGN

E 4.2.1 Background and Rationale

This course separates and makes explicit the decisions that make up an OO analysis and design. It shows how to use the UML notation most effectively.

The rationale of this course is to make the student understand in broad terms the concepts of object oriented systems analysis and design. They should be able to describe the work of the system analysts.

Learning Outcomes

At the end of the course students will be expected to;

- Use the knowledge obtained to analyze the systems;
- Design a new system and offer recommendations

Course Content

- **Object-Oriented Programming:** Introducing the principles of OO design, aims of OO design, OO designs are based on the "real" world, what is an object? distribution of responsibilities, objects, classes, messages and methods, decoupling for flexible software, interfaces and the "implements" relation, types and object specifications, class extension
- **Object-Oriented Analysis And Design:** Aims and overview of process, layering design decisions: abstraction, exposing gaps and inconsistencies: precision, traceable designs: continuity; clear, ready **communication:** a language for design, a vanilla process: development from scratch (brief overview), business modelling: concepts and tasks, systems requirements models, responsibilities and collaborations, persistence, GUI, distribution, coding in an OO language, component-based design (brief overview), robust flexible software, components and interfaces, component kits and architecture, component and re-use culture, patterns (brief overview)
- **Business Modeling And UML Basics:** This section covers techniques of identifying business concepts and tasks, and introduces relevant parts of UML along the way. Static models, objects, types, attributes, snapshot, subtypes, dynamics, use-cases and tasks, event charts, state charts, building a business model, finding a business

model, finding use-cases, connecting usecase and class views, the dictionary, UML notation review, uses of business models, architecture of business process, context of software requirements, basis for component interface definition, documentation style

- **Requirements Modeling:** This section deals with the specification of requirements of a software component, application, or complete system. More modeling patterns and techniques are investigated. System context models, high-level operation specifications, state charts for system models, meaning of "model", how to start abstract and get more detailed, event charts: horizontal and vertical expansion, elaborating models, relating the levels of detail, building a system specification, system context, defining use-case goals, modelling patterns
- **Basic Design:** The key principle of OO design is assigning responsibilities and designing collaborations . Separating core from GUI, persistence and other layers, selection of control objects, designing system operations with messages, decoupling, extensibility, reusability, CRC "cards", dependencies and visibilities, the class dictionary, translation to code (Java, C++ or Smalltalk examples)s

Assessments

- Continuous Assessment 40%
- Final Exam 60%

Prescribed textbooks

1. Schach, Stephen R., Introduction to Object-Oriented Analysis and Design, McGraw Hill 2003

Recommended Text

1. Object-Oriented and Classical Software Engineering. Schach, Stephen R. McGraw Hill 7th Edition 2006
2. Object-Oriented Design and Patterns. Horstmann, Cay S. John Wiley and Sons 2nd Edition 2005
3. Object-Oriented System Development: A Gentle Introduction. Britton, Carol; Doake, Jill. McGraw Hill 2000