Software Analysis and Design (CS3004)

Course Instructor: Nida Munawar
Email Address: nida.munawar@nu.edu.pk

Reference Book: Applying UML and Patterns (An introduction to Object-Oriented Analysis and Design And Iterative Development)

BY Craig Larman

Third Edition

Rules that will be strictly followed:

- All the assignments and projects will be uploaded and submitted via google classroom.
- No assignment will be accepted via email. (Excuses will not be accepted)
- All the quizzes will be announced a week before.
- Missed quiz/assignments will not be retaken. (Excuses will not be accepted)

Announcement:

- Project details and course outline will be shared with you soon via Google classroom.

Tentative Marks distribution:

Assignment:10

Project: 10

Sessional: 30 (15 each)

Final: 50

PS: Absolute Grading scheme will be followed

What you will learn….

- In the course, you will learn about object-oriented Analysis and Design.
- This course will build upon the basics of Java and take you to the next level by covering object-oriented analysis and design.
- You will discover how to create flexible, reusable, and maintainable software by applying object oriented design principles.
- You will learn how to communicate these designs by expressing them in a visual notation known as Unified Modeling Language or UML.
- After MID II focuses on design patterns.
- Design issues and applications can be resolved through design patterns commonly applied by experts.

Course Catalogue - HEC

- Object-Oriented Analysis: Developing the Static Model, Class Diagrams, UML Relationships: Association, Aggregation, Composition, Inheritance. UML Packages, Object Diagrams.
- Object-Oriented Analysis: Developing the Dynamic Model ,Use Case Diagrams, Sequence Diagrams, Collaboration Diagrams, Statechart Diagrams, Advanced States: Substates, Superstates,Activity Diagrams.

Course Goals

- After successful completion of this course students should be able to do analysis of software system. Do modeling using UML and create diagram like use cases, activity, class, sequence, collaboration etc.

Take a minute and think of the projects that you worked on?

- You've probably already worked on some software projects in the past
- Did they have a good design?
- Could the design be done better?
- Was there even a design at all?
- Think of how easy it was to make changes to your code.
- Did a small code change produce a ripple-effect for changes elsewhere in the code?
- Was your code hard to reuse?

Good design isn't just about code.

- It is about being able to express ideas for your software with other developers, other teams, and your clients.
- Having a well-thought design makes your software easier to implement, reduces a need for major changes later or the project and it saves you from headaches down the line.

Consider this scenario.

- You join a project that's been in development for a while.
- You look at the code and become instantly overwhelmed.
- You can't tell what the purpose of the pieces are, things are unorganized, and design documentation is non-existent.
- You don't even know where to begin?????

Software?

- **Computer software**, or simply **software**, is a collection/set of **instructions**, **data** or **programs** that tell the computer how to work.

- A software plays a key role of a mediator between the user and the computer hardware.In the absence of software, a user essentially can’t perform any task on a computer.
- **Types of Software**

1. **Application software**
2. **System software**

What is Analysis and Design?

- **Analysis** emphasizes an *investigation* of the problem and requirements, rather than a solution. For example, if a new online trading system is desired, how will it be used? What are its functions?
- "Analysis" is a broad term, best qualified, as in *requirements analysis* (an investigation of the requirements) or *object-oriented analysis* (an investigation of the domain objects).

What is Analysis and Design?

- **Design** emphasizes a *conceptual solution* (in software and hardware) that fulfills the requirements, rather than its implementation. For example, a description of a database schema and software objects. Design ideas often exclude low-level or "obvious" details obvious to the intended consumers. Ultimately, designs can be implemented, and the implementation (such as code) expresses the true and complete realized design.
- As with analysis, the term is best qualified, as in *object-oriented design* or *database design*.

What is Object-Oriented Analysis and Design?

- During **object-oriented analysis** there is an emphasis on finding and describing the objects or concepts in the problem domain. For example, Bank system(find the objects)
- During **object-oriented design** (or simply, object design) there is an emphasis on defining software objects and how they collaborate to fulfill the requirements. For example, a *Bank_Customer* software object have a
- *AccountNumber* attribute and a *getAccount History* method

Software analysis goal:

- The goal of requirement analysis phase is answer to question: what software must do (and with what constraints)?
- The goal of software analysis phase is answer to question: how system should work?
- The goal of software design phase is answer to question: how system should be implemented?