Introduction to Software Eng.

Divyansh Rajesh Jain

UCDAVIS

What is Software Engineering?

- Engineering “large-scale” software
- Difference between programming and software engineering
    - Maintainability
    - Scalability
    - Reliability

Software Development Lifecycle

- Requirements (This is from user or client)
- **Design**
- Implementation
- Testing
- Deployment
- Maintenance

Important Design Principles

- Modularity and Reusability
- Abstraction
- Design Patterns
    - Battled tested ways to design software!

Challenges in Software Engineering

- Economic Challenges
  - Best quality for lowest cost
- How were these complex problems tackled?

Modularity and Decomposition

- Split functional components away from each other
- Requires “proper” decomposition of functionality

What is Decomposition?

- Split a very complicated problem into smaller, easier problems
- All complex software systems were only possible because of decomposition
    - Example: Compilers

Ex: Task Management System

- Add Users (These are who tasks will be assigned to)
- Add Tasks (Description, Due Date, Assignee)
- Mark Task as Complete
- View Tasks (All tasks, per user)

Thank You!

Next Time: Intro to Structured Development