Software Analysis and Design (CSE 460)

Note: The information below and course outline are subject to modifications and updates.

About this Course
Object-oriented and structured analysis and design; software architecture and design patterns; component-based development; software architecture for mobile and distributed systems; software quality, safety and reliability.

Specific topics covered include:
- Nature of Software Systems and Significance of Software Analysis and Design
- Object Oriented Analysis and Design
- Software Architecture and Design Principles
- Quality Attributes of Software Architecture
- Architecting Mobile Applications
- Architecting SOA and microservices based Web Applications
- Architecting Robotics/Autonomous Systems

Required Prior Knowledge and Skills
- Software life cycle models
- Project management
- Team development environments and methodologies
- Software architectures
- Professional Background: object-oriented programming exposure, basic knowledge of software process modules, class diagrams, experience working on a software development team, experience developing software following a disciplined development process

Learning Outcomes
Learners completing this course will be able to:
- Evaluate software complexity and scale traits in modern software systems.
- Articulate software specification analysis and design concepts with foundations.
- Apply standardized structural and behavioral modeling methods and practices for software systems.
- Demonstrate engineering methodology in analysis and design of a model-based software system.
- Use software design pattern concepts and models in designing a new software system.
· Evaluate and apply appropriate software architecture functional and nonfunctional concepts, quality attributes, and styles in designing a new software system.
· Develop structural and behavioral specifications with advanced features using the Unified Modeling Language (UML), C4 Model for Software Architecture (C4), frameworks and tools.
· Apply appropriate architectural styles in designing and implementing software systems in different application domains including, but not limited to mobile, service-oriented, and autonomous and robotics systems.
· Develop, implement, and test consumer/producer style software systems using software design patterns.

**Major Project**

**Description:** The hands-on projects for this course span across multiple units and apply the fundamentals of design, implementation, and software architecture to demonstrate real world applications. To resemble the industry’s work environment and enhance collaborative learning, some aspects of these projects involve peer feedback and peer evaluation. Specific projects include:

· Eliciting requirements, object identification, modularity and application of Object-Oriented Design patterns in designing and implementing a software project
· Design patterns in designing and implementing a software project
· Software Architecture and Quality Assessment Plan of a software project
· Design and implementation of a mobile application
· Service Oriented Application Design and Implementation Project
· Simple Robotics Simulation Application Tool Project

**Course Content**

**Instruction**

· Video lectures
· Other videos (animations, demos, etc.)
· Readings
· Live sessions (webinars, virtual office hours)

**Assessments**

· Practice activities and quizzes (auto-graded)
· Practice assignments (instructor- or peer-reviewed)
· Team and/or individual project(s) (instructor-graded)
· Midterm or final exam (proctored, auto-and/or instructor-graded)

**Estimated Workload/ Time Commitment Per Week**

15 - 20 hours per week

**Technology Requirements**

**Hardware**

· Computer able to run Java IDE, Visual Studio (to create C# Projects), Android SDK
· Having Mac computer or an access to Mac computer is highly recommended
Software and Other (programs, platforms, services, etc.)

- Java development IDE
- Visual Studio
- XCode/Android Studio
- UML modeling tool such as Astah
- C4 modeling tool such as draw.io

Note: All of these software systems are either open source, free download or can be downloaded through myapps.asu.edu for ASU students free of charge

Course Outline

Unit 1: Nature of Software Systems and Significance of Software Analysis and Design
Module 1: Complex Nature of Modern Software Systems
Module 2: Architecture-Centric Software Development

Unit 2: Object-Oriented Analysis and Design
Module 1: Object-Oriented Design Fundamentals
Module 2: From Requirements to Object
Module 3: Documenting Software Analysis and Design Using Unified Modeling Language (UML)
Module 4: Object-Oriented Analysis and Design Case Study

Unit 3: Software Architecture Preliminaries and Software Design Principles
Module 1: Object-Oriented Design Fundamentals
Module 2: From Requirements to Object
Module 3: Documenting Software Analysis and Design Using Unified Modeling Language (UML)
Module 4: Object-Oriented Analysis and Design Case Study

Unit 4: Software Architecture Fundamentals, Documentation and Evaluation
Module 1: Software Architecture: A Deep Dive
Module 2: Documenting Software Architecture
Module 3: Software Architecture Quality Attributes
Module 4: Evaluation Software Architecture: Architecture Tradeoff Analysis Approach (ATAM)

Unit 5: Architecting Mobile Applications
Module 1: Introduction to Mobile Computing and Special Considerations in Mobile Application Design
Module 2: Model-View-Controller (MVC) Architecture for Mobile Application Development
Module 3: Web Data Integration for Mobile Applications
Module 4: Mobile Application Design and Implementation Case Study
Unit 6: Architecting Service Oriented and Microservices Based Web Applications
Module 1: Introduction to Service Oriented Computing (SOC)
Module 2: Service Oriented Architecture Foundations and SOC Application Design Principles
Module 3: SOC Application Design and Implementation Case Study
Module 4: Microservices Architecture Principles

Unit 7: Architecting Robotics and Autonomous Software Systems
Module 1: Robotics and Autonomous systems: Challenges and Opportunities
Module 2: Software Abstractions and Design Considerations
Module 3: Robotics and Autonomous Systems Design Case Studies

About ASU
Established in Tempe in 1885, Arizona State University (ASU) has developed a new model for the American Research University, creating an institution that is committed to access, excellence and impact.

As the prototype for a New American University, ASU pursues research that contributes to the public good, and ASU assumes major responsibility for the economic, social and cultural vitality of the communities that surround it. Recognizing the university’s groundbreaking initiatives, partnerships, programs and research, U.S. News and World Report has named ASU as the most innovative university all three years it has had the category.

The innovation ranking is due at least in part to a more than 80 percent improvement in ASU’s graduation rate in the past 15 years, the fact that ASU is the fastest-growing research university in the country and the emphasis on inclusion and student success that has led to more than 50 percent of the school’s in-state freshman coming from minority backgrounds.

About Ira A. Fulton Schools of Engineering
Structured around grand challenges and improving the quality of life on a global scale, the Ira A. Fulton Schools of Engineering at Arizona State University integrates traditionally separate disciplines and supports collaborative research in the multidisciplinary areas of biological and health systems; sustainable engineering and the built environment; matter, transport and energy; and computing and decision systems. As the largest engineering program in the United States, students can pursue their educational and career goals through 25 undergraduate degrees or 39 graduate programs and rich experiential education offerings. The Fulton Schools are dedicated to engineering programs that combine a strong core foundation with top faculty and a reputation for graduating students who are aggressively recruited by top companies or become superior candidates for graduate studies in medicine, law, engineering and science.
About the School of Computing, Informatics, & Decision Systems Engineering

The School of Computing, Informatics, and Decision Systems Engineering advances developments and innovation in artificial intelligence, big data, cybersecurity and digital forensics, and software engineering. Our faculty are winning prestigious honors in professional societies, resulting in leadership of renowned research centers in homeland security operational efficiency, data engineering, and cybersecurity and digital forensics. The school's rapid growth of student enrollment isn't limited to the number of students at ASU’s Tempe and Polytechnic campuses as it continues to lead in online education. In addition to the Online Master of Computer Science, the school also offers an Online Bachelor of Science in Software Engineering, and the first four-year, completely online Bachelor of Science in Engineering program in engineering management.

Creator

Janaka Balasooriya, Ph.D.

Janaka Balasooriya joined Arizona State University in 2007. Prior to joining ASU, Balasooriya was a postdoctoral fellow at Missouri University of Science and Technology. With several years of industry experience as a Software Engineer, his research interests span the areas of distributed computing and software engineering, including service-oriented computing, cloud computing, and software testing. Balasooriya has taught courses in Distributed Computing, Mobile Computing, Software Testing, Algorithms and Data Structures, Software Engineering, and Programming Languages. Balasooriya is an ASU Barrett Honors faculty and serves as a faculty honors advisor to CS and CSE students. He is also a program committee member in several premier conferences, including IEEE Service Oriented Computing and IEEE Cloud Computing Conferences since 2007, and serves as an editorial board member of The Services Transactions on Cloud Computing (IJCC).