

**Brooklyn College**  
**Department of Computer and Information Sciences**

**CISC 3171 [30.1] Introduction to Software Engineering**

3 hours; 3 credits

A broad view of software engineering that introduces a variety of software engineering techniques that can be applied to practical software projects. Topics include: process models, software specification, software design, software development methods and tools, verification and validation, reliability, and human factors.

**Syllabus**

1. Software Engineering: A preview. (1 week)  
History of SE, role of the software engineer, the software life cycle, relationship of SE to programming languages, operating systems, data bases, artificial intelligence, and theory of computer science.
2. Software: It's Nature and Qualities (1 week)  
Classification of software qualities, quality requirements in different application areas, measurement of quality.
3. Software Engineering Principles (2 weeks)  
Rigor and formality, separation of concerns, modularity, abstraction, anticipation of change, generality, incrementability.
4. Software Design (1 week)  
Software design activity and objectives, modularization techniques, object-oriented design, handling anomalies, a design case study, concurrent software.
5. The Software Production Process (2 weeks)  
Software production process models, waterfall model, evolutionary model, transformation model, spiral model, model assessment, case studies, organizing the process.
6. Software Specification (1 week)  
Specification usage and qualities, specification styles, specification verification, operation vs. descriptive specifications, specification in practice.
7. Software Verification (1 week)  
Goals and requirements, approaches, testing, analysis, symbolic execution, debugging, verifying other software properties.
8. Management of Software Engineering (1 week)  
Management functions, project planning, project control, organization, risk management.
9. Software Engineering Tools and Environments (2 weeks)  
Historical evolution, classification, representative tools, role of programming

languages, sample tools and environments, future scenario.

10. Testing Programs and Systems (2 weeks)  
software faults and failures, testing issues, unit testing, integration testing, performance testing, reliability, fault tolerance, testing safety-critical systems, real-time examples.

### **Primary Text:**

Fundamentals of Software Engineering, Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli, Prentice-Hall, 1991, ISBN: 0-13-820432-2, 573 pages.

### **Other References**

Software Engineering: The Production of Quality Software by Shari Lawrence Pfleeger, 2nd Edition, Macmillan, 1991, ISBN: 0-02-395115-X.

Software Engineering with Abstractions by Valdis Berzins and Luqi, Addison Wesley, 1991, ISBN: 0-201-08004-4, 592 pages.

Classic and Object-Oriented Software Engineering, 3rd Edition, by Stephen R. Schach, Richard D. Irwin, Inc. ISBN: 0-256-18298-1, 1996.

Experimentation in Software Engineering, by Claes Wohlin, Per Runeson, Martin Host, Magnus C. Ohlsson, Bjorn Regnell and Anders Wesslen, Lund University, Sweden. Kluwer Publishers, 1999, ISBN: 0-7923-8682-5, 224 pages.

Software Engineering, 6<sup>th</sup> Edition, by Ian Sommerville, Addison-Wesley, 2001, ISBN: 0-201-39815-X, 693 pages.

Software Engineering: Theory and Practice, 2<sup>nd</sup> edition, 2001, By Shari Lawrence Pfleeger, Prentice Hall; ISBN: 0130290491 659 pages.