

1. Course number and name

CptS 422: Software Engineering Principles II

2. Credits and contact hours

3 credits, 3 lecture hours

3. Instructor's or course coordinator's name

Haipeng Cai

4. Textbook, title, author, and year

P. Jorgensen. 2013. *Software Testing, A Craftman's Approach* (4th ed.). CRC Press. ISBN-13: 978-1466560680.

I. Burnstein. 2003. *Practical Software Testing*. Springer. ISBN-13: 978-0387951317.

I. Bashir and A. Goel. 1999. *Testing Object-oriented Software: Life Cycle Solutions*. Springer. ISBN-13: 978-0387988962.

Robert V. Binder. 1999. *Testing Object-Oriented Systems – Models, Patterns, and Tools*, Addison-Wesley. ISBN-13: 078-5342809381.

I. Sommerville. *Software Engineering* (9th ed.). Addison Wesley. ISBN-13: 978-0137035151.

5. Specific course information

a. *Catalog description*: Dependable software systems; software verification and validation, testing; CASE environments; software management and evolution.

b. *Prerequisites or corequisites*: Certified major in CptS, CE, EE, or SE, have finished CPT S 321 with a C or better or CPT S 323 with a C or better; CPT S 322 with a C or better;

6. Specific goals for the course

By the end of the course, students will be able to

- Describe different test levels and testing objectives (3a, 3e).
- Apply test methods for the different phases of development and life cycle of the software (1a, 1b, 2c, 2e, 5b, 5d, 5f, 5g, 6b).
- Identify coverage and acceptance criteria for the tests based on the programming activities and the phase of development (1b, 2c, 6c, 6d).
- Assess the quality and reliability of the software system (6a, 6b, 6c, 6d).
- Plan and appropriately document for software testing activities (3a, 3c, 7a, 7b, 7d, 7e).

7. Brief list of topics to be covered

- Fundamental software testing concepts
- Different test levels (e.g., unit testing, integration testing, system testing)
- Objectives of testing: acceptance testing, installation testing, alpha/beta testing, performance testing, etc.

- Testing techniques (e.g., black box, white box, mutation testing, etc.)
- Reliability evaluation
- Test-related measures (e.g., fault density, mutation score)
- Test planning and documentation