BAKER COLLEGE
STUDENT LEARNING OUTCOMES

CS4510A Software Testing
3 Semester Credit Hours

Student Learning Outcomes and Enabling Objectives

1. Distinguish among the different types of testing.
    a. Discuss the issues involved in the testing of object–oriented programs.
    b. Describe different unit testing frameworks.
    c. Compare various unit testing frameworks.
    d. Contrast different mocking frameworks.
    e. Describe different dependency injection frameworks.
    f. Review techniques for integration testing.
    g. Explain how to write an automated integration test.
    h. Differentiate "Black Box" testing and "White Box" testing
2. Discover the role that unit tests play in the validation of software.
    a. Explain the classical approach to software development and testing.
    b. Describe the purpose of Test-Driven Development (TDD) as a design methodology.
    c. Create a set of tests for a code segment.
    d. Apply the TDD approach for software development and testing.
    e. Explain the purpose of refactoring.
3. Apply techniques for mocking external resources.
    a. Explain the purpose of mocking external resources.
    b. Generate mock tests for a medium-size code segment.
    c. Differentiate a mock test from functional and integration tests.
4. Construct good regression tests and automate them.
    a. Explain potential causes of defects during software changes.
    b. Describe the purpose of regression testing.
    c. Formulate a policy for regular regression testing.
    d. Identify best practices for automation analysis.
    e. Build regression test cases.
    f. Analyze pros and cons of various automated testing tools available for purchase.
5. Use a defect tracking tool to manage software defects on a small software project.
    a. Explain the purpose of defect tracking tools.

b. Explore important features (reporting, assigning, progressing, storage and retrieval) of various defect tracking tools.
c. Explore how a defect tracking tool is typically used to manage software defects.

Big Ideas and Essential Questions

Big Ideas

- Design practices for validating software, ensuring reliability, facilitating maintainability and facilitating performance optimization
- Tools and Frameworks for testing software
- Designing testable software
- Software testing best practices
- Evolution of software testing practices and standards

Essential Questions

1. What factors determine quality software?
2. How can software programs be designed and developed to maximize the users' experience?
3. How can software programs be effectively tested, maintained and upgraded to meet the evolving users' expectations?
4. How do you continuously improve the quality of software programs through the use of state-of-the-art testing and validation best practices?

These SLOs are not approved for experiential credit.

Effective: Fall 2023