CISC 3120

C26a: Testing UI and Web Service

Hui Chen

Department of Computer & Information Science

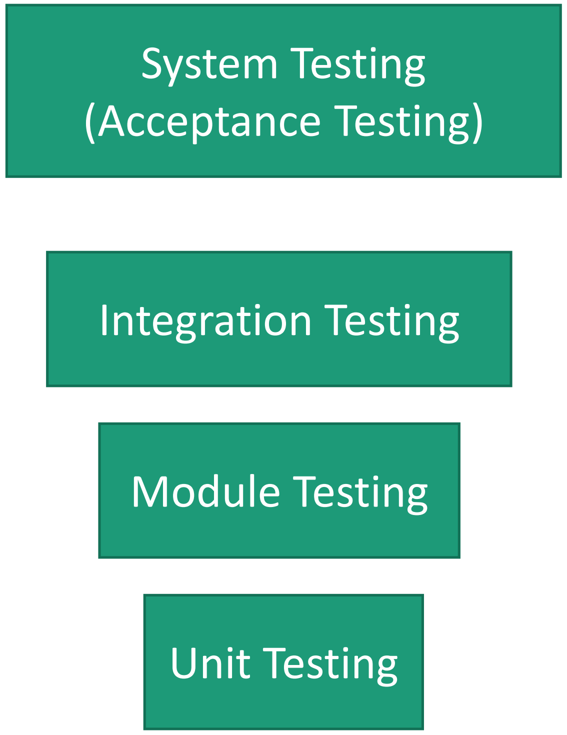
CUNY Brooklyn College

Outline

- Recap
  - Your reflection?
- Test-driven development
- Automatic testing for user interface
- Testing Web services with the Spring Framework

Different Types of Tests

- System testing (acceptance testing): if the integrated program meets its specification
- Integration testing: interfaces between units have consistent assumption and communicate correctly
- Module testing: tests across individual units (e.g., across classes)
- Unit testing: single method does what was expected (e.g., within a single class)



Approaches to Testing

- Ad-hoc and “eye-ball” testing
- Formal approach
    - Automated or semiautomated testing

Test-driven Development

- Testing is an integral part of software development
    - Software development can even begin with testing
    - Test-driven development
        - Design a new feature beginning or along with design tests
        - Implement the design
        - Run tests
        - Refactor the design and revise and the implementation
- Help learners as well
    - Examples:
        - What should be the specification?
        - What should be the input, and what should be the output?
        - Does my code work? Start from small?

Design for Testing

- Add methods to aid testing
  - Some methods may be added just for testing purpose.

"But, you don't want to break the good design just for unit tests!"

"Testable code tends to be good code, and vice versa."

Questions

- Concept of software verification and testing
- Approaches to testing
  - Formal or ad hoc?
- Concept of test-driven development

Testing User Interface

- Applications can have sophisticated user interface
- Ad-hoc testing
  - Run the programs, and observe how it behaves
    - Simple, fast, does not require additional knowledge; however, inaccurate, costs more and more time later in the project
- Automated UI testing
  - Emulate user interaction with the application
    - Require support from some testing framework to emulate user interaction and capture output;
    - Building and maintaining tests for dynamic and evolving user interfaces can be very high
    - Benefit more when the application becomes stable

JavaFX: Automated UI Testing

- A few test frameworks
  - Jemmy from OpenJDK
    - http://hg.openjdk.java.net/code-tools/jemmy/v3/
  - TestFX
    - https://testfx.github.io/src/

TestFX: Example

- Add a test in the AddressAutoFillerFX example

Questions?

- User interface testing
  - Automated user interface testing

Testing Web Services

- The Spring framework supports automated testing
  - Unit testing
  - Integrated testing

Questions

- Unit and integration testing in the Spring framework

Further Reading

- Testing for JavaFX via TestFX
  - https://github.com/TestFX/TestFX/wiki
- Testing in Spring Framework
  - https://docs.spring.io/spring/docs/current/spring-framework-reference/testing.html