EECS 293
Software Craftsmanship
2019 Fall Semester

Programming Assignment 12

Due at the beginning of your discussion session on

April 16-19, 2019

No late assignment will be accepted after April 30, 2019

Reading

- Section 6.2, "Present a consistent level of abstraction in the class interface" in Code Complete
- Section 6.3 in Code Complete
- Items 1, 2, 10, 18, 19, 20 in Effective Java
- Section 19.6 in Code Complete
- Module "Routine Names" on canvas

Grading Guidelines

Points will be deducted if code and branch coverage is incomplete. You can omit coverage of methods that are automatically generated and of assertions. An automatic C (or less) is triggered by:

- Any routine with complexity greater than 4,
- Any substantially repeated piece of code, or by
- Improperly named routines.

Programming

Make any changes to the design as required after the review and this week's lecture. Implement your design:

- Interfaces and classes
- Methods (according to your pseudo-code, if any)
- Error-handling

Test cases are required to validate and improve your design. No test cases are required for automatically generated methods. Focus on unit tests, plus a single stress test.

General Considerations

After Programming Assignment 8, your code should have an extensive unit test suite. Your code should have a reasonable number of comments, but documentation is going to be the topic of the next assignment. As a general guideline, comments should be similar to those accepted in EECS 132.

Submission

Create a repository called typecheck.git. Make small regular commits. Push your design document, revised code, and test cases on the git repository.