The beginning of wisdom for a programmer is to recognize the difference between getting his program to work and getting it right. A program which does not work is undoubtedly wrong; but a program which does work is not necessarily right. It may still be wrong because it is hard to understand; or because it is hard to maintain as program requirements change; or because its structure is different from the structure of the problem; or because we cannot be sure that it does indeed work. (Jackson, *Principles of Program Design*, 1975)

### 1 Instructor
**Name** Joel Coffman  
**Phone** 240 228-0544 or 443 778-0544 (office)  
**Email** joel.coffman@jhu.edu (preferred)  
joel.coffman@jhuapl.edu  
**Website** [http://pages.jh.edu/~jcoffma2](http://pages.jh.edu/~jcoffma2)

The preferred way to reach me is via email. I make every effort to respond to email within 24 hours although this time frame may not always be feasible. *Please be aware that I rarely check or respond to email on Sundays.*

### 2 Course Description
Fundamental software engineering techniques and methodologies commonly used during software development are studied. Topics include various life cycle models, project planning and estimation, requirements analysis, program design, construction, testing, maintenance and implementation, software measurement, and software quality. Emphasized are structured and object-oriented analysis and design techniques, use of process and data models, modular principles of software design, and a systematic approach to testing and debugging. The importance of problem specification, programming style, periodic reviews, documentation, thorough testing, and ease of maintenance are covered.

### 3 Course Goals
The goal of this course is to provide an overview of the fundamental concepts of software management as well as the design and implementation of software systems. Various organizational structures and life cycle processes are covered. This course provides hands-on software development activities that emphasize the importance of design, code review, testing, and maintenance.

### 4 Course Objectives
By the conclusion of this course, students are expected to
- have a thorough understanding of software management fundamentals, particularly software development processes;  
- use version control to manage changes to software and to deliverables;  
- create software abstractions that facilitate testing;  
- write thorough unit and integration tests for software; and  
- document the end products developed along with software (e.g., requirements, source code, and end-user documentation).
5 Course Structure
This course is divided into separate modules. Each module typically comprises reading to prepare for the class meeting, in-class discussions and lecture, and a homework assignment.

6 Textbook
Required

7 Technical Requirements
Proficiency in a high-level programming language and familiarity with object-oriented design is essential. The class project will likely require programming in Java although other languages may be used in some cases.

8 Student Coursework Requirements
8.1 Grading
Assignments are due by the date listed with the assignment. If you have a legitimate reason for turning something in late, you must notify me ahead of time; I can be flexible when appropriate. Otherwise, late assignments either will not be accepted or, if the situation dictates, will be penalized 10% for each day that the assignment is late.

Final grades will be determined using the following weighting and standard grading scale:

<table>
<thead>
<tr>
<th>Item</th>
<th>%</th>
<th>Grade</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekly Quizzes</td>
<td>40</td>
<td>A</td>
<td>90–100</td>
</tr>
<tr>
<td>Project / Assignments</td>
<td>40</td>
<td>B</td>
<td>80–89</td>
</tr>
<tr>
<td>Final Exam</td>
<td>20</td>
<td>C</td>
<td>70–79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F</td>
<td>&lt;70</td>
</tr>
</tbody>
</table>

± may be applied to final grades at the discretion of the instructor.

8.2 Project
Students will contribute to an ongoing class project that spans individual class offerings. The class project has an open source license and may be released publicly once it reaches sufficient maturity. Please speak to me if you cannot (or are unwilling to) contribute to an open source project.

8.3 Absences
Quizzes missed due to excused absences (e.g., work-related travel) cannot be made up, but students have the option, if it benefits them, of counting their grade on the final as a replacement for missed quizzes.

9 Policies and Guidelines
9.1 Academic Misconduct Policy
The University policy on academic integrity is summarized in the Engineering and Applied Science Programs for Professionals catalog. The consequences of violating this policy are detailed on the Whiting School of Engineering’s website: http://ep.jhu.edu/wseacademicmisconductviolation.