# CS 580.1/483.1 – Software Specification and Analysis - 3 Credit Hours

**Textbook:**

*Using Z: Specification, Refinement, and Proof*, by Woodcock, J., and Davies, J. PH, 1996.

**Alternate References:**

*Modeling Reactive Systems With Statecharts: The Statemate Approach*, by Harel, D., Politi, M., McGraw Hill, 1998.
*Specification Case Studies (2nd Ed)*, by Hayes, I., PH, 1993.
*Z Reference Manual, Spivey, M. (available on the web)*
*Formal Methods for Real-Time Computing,* Edited by Heitmeyer, C. and Mandrioli, Wiley, 1996.
*Application of Formal Methods*, by Hinchey, M.G., and Bowen, J.P., PH, 1995.
*Software Engineering with B*, by Wordsworth, J.B., Addison-Wesley, 1997.
*Other reference material may be presented in class (for which the student is responsible).*

**Textbook Coverage and Supplemental:**

Using Z:
Chapters 1 – Introduction
Chapters 2 – 3 – Propositional and Predicate logic
Chapters 4 – Equality and Definite Description
Chapters 5 – 6 Sets and Definitions
Chapters 7– 8 Relations and Functions
Chapters 9 – 10 Sequences and Free Types
Chapters 11 – 12 Schema and Schema Operators
Chapters 13 – 14 Promotion and Preconditions
Chapters 15 – 23 Examples
Specification Case Studies:
Part 1 Tutorials, and Part 2 Software Engineering
Modeling Reactive Systems With Statecharts:
Supplemental readings as required by the instructor.

**Course Objective and Description:**

Introduction to formal methods used in software engineering: Formal mechanisms for specifying, validating and verifying the correctness, reliability and efficiency of software systems. The course will first introduce the broad area of formal methods and abstraction. The class will then focus on developing a working knowledge for using Z. A project will require the use of the Z language in developing a formal specification of a particular real world requirements specification. The Z specifications will then be converted to Statecharts using the Magnum Statemate tool. Outside readings will be assigned that report on a range of independent experiments devoted to broadening the link between theory and practice including case studies. Prerequisites include knowledge of modern programming languages, data structures, algorithms and discrete mathematics.

**Topical Outline:**

| *Topic* | *Reference (Chap)* | *Lectures* |
|---|---|---|
| 1. Course introduction and objectives including an overview of important topics and project initiation. Why are formal methods important including both algebraic and model based specification techniques. | 1 | 9 |
| 2. Logic, sets and relations | 2 - 4 | 4 |
| 3. Sets and Relations | 5 - 10 | 6 |
| 4. Introduce the Schema Language | 11 - 14 | 6 |
| 5. Woodcock: Example Systems *Refinement* | 16 - 19 | 6 |
| 6. Hayes: Specification Case Studies (Tutorials and Software Engineering) | Parts I and II | 6 |
| 7. Woodcock: Case Studies | 15, 20-23 | 6 |
| Total (1 or probably 2 lectures will be used for in class exam) | 28 | |

**Grading Policy:**

- There will be (closed book/notes) 1 (possibly 2) regular class period exams plus a comprehensive final exam.
- There will be 8 homework assignments. Quizzes are possible but not likely (unless I see that people are not keeping up). Study questions for the material to be covered by the exam will be provided.
- Extra credit for outstanding contributions to projects as well as an extra credit question (or two) on the exams will be awarded. Don't be surprised.
- This class is a lot of work. If you have a full time job and more than one other Computer Science class then I suggest that you drop either this class (or the other) now and not later when it may hurt you. *In fact, if you do not come and talk to me in such a case to get my permission then you unofficially do not have my permission to take this class.* This is partly because it is not fair to your team mates if you never have the time to either meet with them or to cover your end of the deal with respect to the project.

**Grading Distribution and Criteria**

| | | |
|---|---|---|
| Mid-term (In-class) examination | | 20% (Undergraduates 25%) |
| 1 Final (Take-home) examination | | 25% (Undergraduates 30%) |
| 1 Term project | | 25% (times the pe-factor) |
| • Deliverables | | |
| • Proj. Plan +Status reports (Emailed wkly) | (10%) | |
| • Software requirements spec. | (15%) | |
| • Presentation | (05%) | |
| • Peer evaluation | | Final project grade times pe-factor |
| 1 Term paper (for graduate students only) | | 10% |
| Homework and Quizzes | | 10% |
| Discretionary | | 10% |
| Total | | 100% |

Final Grade Grad = 0.10*HQavg + 0.25*PJavg*(PENavg/100) + 0.10*TPpr + 0.20*Ex1 + 0.25*Final + 0.10*Discr
Final Grade Ugrad = 0.10*HQavg + 0.25*PJavg*(PENavg/100) + 0.25*Ex1 + 0.30*Final + 0.10*Discr
PENavg = ((PE1 + PE2)/2)*25 [the 25 normalizes the 0-4 ranking to 100%]

All grades are based on a decade scale from 0-100 as follows:

| | | |
|---|---|---|
| 90-100 | = A | A linear shift (known as curving) may be applied to the final grade averages |
| 80-89 | = B | as a one-time scale at instructors discretion. Plus and minus grades may be |
| 70-79 | = C | given out on the final course grade at the instructors discretion. Note, a |
| 60-69 | = D | grade of 'C' (or lower) is considered failing for graduate students. |
| < 60 | = F | |

**Student Peer Evaluation of Project Participation**

Ratings (0-4): 4 = excellent (A), 3 = good (B), 2 = fair (c), 1 = poor (D), 0 = completely failed

Evaluation Categories:

1. Meetings attendance. and timeliness (e.g., delivers work when promised).
2. Participation (e.g., contributes ideas, direction and takes responsibility).
3. Quality of personal deliverables (e.g., above average, average, sloppy, sloppy and no work).
4. Initiative (e.g., offers ideas and takes action, tries to solve problems, researches alternatives).
5. Responsiveness (e.g., to emails, phone calls and personal communications and deadlines).

## General Conduct Policies and Grading Standards

**Policy On Discretionary Points**

Discretionary points, based on your conduct and participation in the class, will be assigned. Actually, my mission is your success. I am not here to defeat you but rather to help propel you to the next level. However, disorderly or disruptive conduct (e.g., asking questions by raising your hand is reasonable and encouraged, but insisting I answer a question that I have acknowledged and have either deferred to a later date or that I deem unimportant to the goals of the class is not), lack of attendance (especially chronic) without reasonable justification (work related absence is tolerable up to a point) are both examples where the student should expect to lose discretionary credit. As a courtesy, you should try to inform me by email that you will not be coming to class. I expect to be treated as a professional on a professional level (i.e. as you would treat your employer). If you have a problem or are confused with something I am doing or are asking you to do in the context of this class then lets try to resolve it as professionals face to face. I can not return phone calls between your meetings at work. In fact, most of the time its more convenient for me to answer emails. But that is not always the case and people too busy to see me (or call me) during my office hours or otherwise (I am very accessible almost anytime) are really too busy to be taking this course. I say this because its sometimes very inefficient to work problems out completely using email. All of this, and in this general area of conduct, falls under the heading of discretionary credit. Students should expect to be judged accordingly.

**Policy on Missed Exams**

Absolutely no makeup exams are given without prior authorization or written proof (or its equivalent) that the student was prevented from participating. Unexcused missed exams result in a grade of zero for the exam. Excused absences from exams include: (1) personal emergencies, (2) work-related (with confirmation letter). Taking an exam early may be possible for *unavoidable*, planned absences ans is not possible for the final (due to University regulations).

**Cheating on Exams or Homework**

Absolutely no cheating on exams or projects and homework will be tolerated. Students are encouraged to discuss concepts for homework (individually and in class); however, each student is expected to develop his or her own solutions. For further details on academic honesty the student is referred to the Student Handbook.

**Class Attendance**

Class attendance is not required (but may count towards your total grade in terms of the discretionary points you are assigned), and you are 100% responsible for all material (and announcements) presented in class. A class attendance sheet will be passed out at the beginning of each class and you are required to initial it.

**Late Drop**

Dropping of a class after the deadline listed in the class schedule is governed by departmental and college policy. The student must show documented evidence supporting reasons for a request to drop a class after the deadline. Each request is considered on an individual basis for determining acceptance.

**Office Hours:**

Typically I will be available after the class meets, or by appointment. Official office hours for this are posted at my office or check my home page at URL: http://www.eecs.wsu.edu/~sheldon. Other modes of communication with me will also be announced -- e-mail (preferred), voice-mail, etc.

**Standards and Grading Criteria for Computer Projects and/or Homework:**

- Homework must be clearly presented and complete. Do not turn in scratch work (or multiple attempts).
- Homework that is not software will not be accepted after the due date.
- Homework is due at the beginning of class (not accepted later) on the due date.
- Your solutions must be clear, effectively organized, and effectively presented. Show the major logical steps in achieving a solution. Do not turn in pages with irrelevant scratch work or marked-through work.
- Questions about solutions to problems are encouraged, but may have to be handled individually since class time is at a premium.
- Check the course web page for project requirements (and schedule), documentation standards, and tailoring guidelines for the IEEE standards. Example artifacts from prior class projects are available in the library.