CIS453: Software Specification and Design (3 credits)

(MW 345-505PM, Shemin Auditorium)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| I**nstructor** | Joe |  |  | |
|  |  |  |  | |
| **Office Hours** | Tue.9-10AM, Thu.11AM-12PM |  |  |

**Grading Assistant - TBD**

## Prerequisite / Co-requisite: CIS351 or CSE382

## Course Description: Software engineering process models; Software requirements analysis, including object-oriented methodology; Behavioral and nonbehavioral requirements; Development of specification documents; Unified Modeling Language; High-level design and software architecture. Team projects.

### Additional Course Description: Software Ethics; Design Patterns

## Bibliography/ Texts / Supplies – Required:

Software Engineering, (10th Edition), Ian Summerville, Pearson, ISBN: 978-0133943030

Systems Analysis and Design: An Object-Oriented Approach **with UML** (6th Edition), by A. Dennis, B.H. Wixom, and D. Tegarden, Pub: Wiley, ISBN: 978-1119561217

**Learning Approach for each topic:**

1. Read pages in textbook (individual)
2. Attend class discussion/lecture on topic
3. Attempt class exercise (with a classmate)
4. Attend discussion of class exercise
5. Homework assignment (individual)
6. Read feedback on assignment
7. Discuss with TA, if needed
8. Work with team-mates on project (team of four)

***Course Schedule:*** Week, topic for the week, and required reading (pages and chapter numbers are from the 6th edition of textbook) are in the appropriate columns below. Read these pages in the textbook, even if they are not all discussed in class.

[“HWn”: individual assignments; “WRn”: weekly reports for team projects.]

| Class dates | Topic | Required Reading | Homework  Dates |
| --- | --- | --- | --- |
| Aug. 30, Sep. 1 | Systems Development; Software Engineering Approaches;  The unified process & UML | 1-40 [ch1 & appendix] | None |
| Sep. 8, 13 | Requirements Determination | 95-125 [ch3] | HW1: 9/20 |
| Sep. 15, 20, 22 | Use-case diagrams & descriptions, Activity diagrams | 126-168 [ch4] | WR1: 9/25  HW2: 9/27 |
| Sep. 27, 29, Oct. 4 | Structural modeling | 169-210 [ch5] | WR2: 10/2  HW3: 10/4  WR3: 10/9  HW4: 10/11 |
| Oct. 6, 11 | Behavioral modeling | 211-252 [ch6] | WR4: 10/16  HW5: 10/18 |
|  |  |  | Proj.1: 10/23 |
| Oct. 13 | Elements of Design; Packages | 255-286 [ch7] |  |
| Oct. 18, 20, 25 | Interface design | 364-413 [ch10] | HW6: 11/1  WR5: 11/13 |
| Oct. 27, Nov. 1, 3 | Class and Method Design | 287-319 [ch8] | HW7: 11/8  HW8: 11/15  WR6: 11/20  WR7: 12/4 |
| Nov. 8 | Data Design (Sanup Araballi) | 320-363 [ch9] |  |
| Nov. 10, 15 | SQL and No-SQL databases |  |  |
| Nov. 17 | Web Development (Subodh Kalia) |  |  |
| Nov. 29 | Project planning & management | 41-92 [ch2] |  |
| Dec. 1 | Cybersecurity in database design (Prof. Yuzhe Tang) |  |  |
| Dec. 6 | Software Architecture and Design Patterns |  |  |
| Dec. 8 | Working in Teams; Communication and Conflict Management; Software ethics |  |  |
|  |  |  | Proj.2: 12/11 |

**Individual Deliverables (by noon on the indicated date):**

* 9/20 (HW1): Suggest a topic for an individual software development project—different from the team project. [Examples: an inventory management system for some organization; software to facilitate some task (e.g., admission application processing) at the University; traffic management or control (cars/trucks/boats); an Uber-style business (where services are provided by a large network of people); an app that would make life easier in some task.] Provide a Requirements Definition (as in Fig.3-11, p.118). **Quiz 1 during Sept. 27-29.**
* 9/27 (HW2): Sketch a use case diagram (as in Fig. 4-4 and 4-6) for the individual project; describe two of the principal use cases in detail (as in Fig.4-17 and 4-18). **Quiz 2 during October 4-6.**
* 10/4 (HW3): Sketch CRC cards (as in Fig. 5-6) for two of the main classes in the individual project. **Quiz 3 during the week of October 11-13.**
* 10/11 (HW4): Sketch the class diagram (as in Fig. 5-23) for the individual project.
* 10/18 (HW5): Sketch sequence diagrams (as in Fig. 6-9) for the individual project. **Quiz 4 during the week of October 18-20.**
* 11/1 (HW6): Describe the external user interfaces for the individual project.

**Quiz 5 during November 8-10.**

* 11/8 (HW7): Describe the most important class in the individual project. Describe its interactions with other classes (data exchange, calling methods, and sequencing). Describe the content and format of data, and list the methods in the class. **Quiz 6 during November 15-17.**
* 11/15 (HW8): Specify all methods of the class (chosen in HW7), following the templates in Fig. 8-11, and Fig. 8-15, including algorithm descriptions (as in Fig. 8-18) for each non-trivial method. **Quiz 7 during November 29-December 1.**

***Team project topics must differ from those of individual homework assignments.***

**Team Deliverables: Due on Saturdays, by midnight.**

(An update will be accepted on Sunday without penalty, if it changes/adds no more than 20% to Saturday’s submission. Else, grade penalties apply, @10% per day.)

**Requirements Team Project:**

9/25: Weekly report 1: project description and requirements.

10/2: Weekly report 2: use case diagram, and use case descriptions.

10/9: Weekly report 3: CRC cards and class diagram.

10/16: Weekly report 4: behavioral descriptions for all classes.

10/23: Complete Software Requirements Document including all of the above.

10/24: Feedback on team project.

**Design Team Project:**

11/13: Weekly report 5: External Interface Descriptions (if your system interacts with hardware or sensors, describe exactly how these interactions occur, including data formats and any handshake protocols).

11/20: Weekly report 6: Detailed descriptions of all classes and their interactions (as in HW7).

12/4: Weekly report 7: Method Descriptions (as in HW8).

12/11: Complete Software Design Document including all of the above, and detailed descriptions of any databases and non-trivial data structures needed for the project

12/12: Feedback on team project.

***Rubrics for Team Project Grading:***

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Poor*** | ***Moderate*** | ***Good*** |
| Weekly progress reports | <75% submitted and adequate: 0.0 | 75% submitted and adequate: 0.5 | All submitted & adequate: 1.0 |
| Completeness | Less than half of required elements present: 0.0 | Most required elements present: 0.5 | All required elements present: 1.0 |
| Consistency | Multiple inconsistent elements: 0.0 | Mostly consistent: 0.5 | Completely consistent: 1.0 |
| Clarity | Difficult to read and understand: 0.0 | Moderately easy to understand: 0.5 | Easy to understand: 1.0 |
| Bonus |  |  | Discussion of choices made, motivation and alternatives: 0.5 |

**Penalties and bonuses for teamwork and unequal contributions:**

* Weekly reports should indicate if any team-members did not participate in that week’s work.
* Confidential teamwork feedback is expected after the first and second projects.
* Each student’s contribution to the project will be estimated from the comments of the other students in the team, and used to determine the student’s project grade.

## Grading: Irrespective of scores, the final course grade will be F if a majority of your team-mates in either project say that you have contributed less than 10% to the project, or if plagiarism is detected in any of your submissions. [For example, even if you get perfect scores in the first project and all other work, your course grade will be an F if you do nothing for the second project.]

* 10%: Class exercises
* 20%: 8 individual HW assignments, equally weighted
* 20%: 7 quizzes, equally weighted
* 30%: Software requirements team project
* 20%: Design team project

**Each score will be on a 0-4 scale, with the weighted average rounded as follows to obtain the final grade--if your team-mates in both projects say that you have contributed sufficiently, and if you have not violated plagiarism norms.**

|  |  |
| --- | --- |
| Grades | Grade range obtained as the weighted average of quizzes and projects |
| A | >3.66 |
| A- | >3.33 |
| B+ | >2.99 |
| B | >2.66 |
| B- | >2.33 |
| C+ | >1.99 |
| C | >1.66 |
| F | 0.0-1.66 |

## Active participation (e.g., asking/answering questions) in classes can provide a bonus of 0.1 (added to the final weighted average) which may help to nudge your grade upwards.

## Course Specific Policies:

* Missed quizzes, assignments, and class exercises will be assigned zero scores.
* Late work will be penalized at the rate of 10% per day.

## Additional Information:

Team project grade scores will be based on the quality of the deliverable, and the contribution of an individual student to the team, as documented in weekly blogs submitted by each team, and confidential feedback. Teams will be assigned quasi-randomly; alert the instructor in advance if there are specific people with whom you do not get along or have had prior conflicts.

## \*Academic Integrity Policy

The course grade will be an F for any violation of academic integrity, and reported to the College and University.

Syracuse University’s Academic Integrity Policy reflects the high value that we, as a university community, place on honesty in academic work. The policy defines our expectations for academic honesty and holds students accountable for the integrity of all work they submit. Students should understand that it is their responsibility to learn about course-specific expectations, as well as about university-wide academic integrity expectations. The policy governs appropriate citation and use of sources, the integrity of work submitted in exams and assignments, and the veracity of signatures on attendance sheets and other verification of participation in class activities. The policy also prohibits students from submitting the same work in more than one class without receiving written authorization in advance from both instructors. Under the policy, students found in violation are subject to grade sanctions determined by the course instructor and non-grade sanctions determined by the School or College where the course is offered as described in the Violation and Sanction Classification Rubric. Syracuse University students are required to read an online summary of the University’s academic integrity expectations and provide an electronic signature agreeing to abide by them twice a year during pre-term check- in on MySlice.

The Violation and Sanction Classification Rubric establishes recommended guidelines for the determination of grade penalties by faculty and instructors, while also giving them discretion to select the grade penalty they believe most suitable, including course failure, regardless of violation level. Any established violation in this course may result in course failure regardless of violation level.

In every assignment or document submitted by an individual or a team, there must be a section titled “References” where students must mention all tools, templates, papers and other documents on which they relied, except for the class notes, textbook and associated resources. Failure to mention this will be assumed to imply plagiarism.

Occasionally, for reasons such as illness or employment-related travel, a student may be permitted to take a quiz (or other classwork) away from the classroom. In such cases, there should not be any communication with other students (regarding the quiz or classwork) until after all students have completed submission of the same. Violation of this expectation will be considered to be an instance of academic dishonesty.

All students contributing to a team submission will be considered responsible for academic integrity violations associated with that submission. Hence, please ensure that your team-mates uphold all academic integrity expectations.

## Student Academic Work Policy

SU policy on student academic work may be found at:

<http://coursecatalog.syr.edu/content.php?catoid=3&navoid=270#Student_Academic_Work>

St**udent work prepared for this course may be used for educational purposes** during this semester and later semesters, as well as in assisting ABET and other evaluations of our undergraduate programs. You grant permission to have your work used in this manner by registering for, and by continuing to be enrolled in this course.

## Disability-Related Accommodations

If you believe that you need accommodations for a disability, please contact the Center for Disability Services (CDS), <http://disabilityservices.syr.edu>, located in Suite 303 of 804 University Avenue, or call (315) 443-4498, TDD: (315) 443-1371 for an appointment to discuss your needs and the process for requesting accommodations. CDS is responsible for coordinating disability-related accommodations and will issue students with documented Disabilities Accommodation Authorization Letters, as appropriate. Since accommodations may require early planning and generally are not provided retroactively, please contact CDS as soon as possible.

### Diversity and Disability:

Syracuse University values diversity and inclusion; we are committed to a climate of mutual respect and full participation. My goal is to create learning environments that are useable, equitable, inclusive and welcoming. If there are aspects of the instruction or design of this course that result in barriers to your inclusion or accurate assessment or achievement, I invite any student to meet with me to discuss additional strategies beyond accommodations that may be helpful to your success.

## Religious Observances Notification and Policy

SU religious observances notification and policy, found at <http://hendricks.syr.edu/spiritual-life/index.html>, recognizes the diversity of faiths represented among the campus community and protects the rights of students, faculty, and staff to observe religious holidays according to their tradition. Under the policy, students are provided an opportunity to make up any examination, study, or work requirements that may be missed due to a religious observance provided they notify their instructors before the end of the second week of classes for regular session classes and by the submission deadline for flexibly formatted classes.

For fall and spring semesters, an online notification process is available for students in **My Slice /** **StudentServices / Enrollment / MyReligiousObservances / Add a Notification.**  Instructors may access a list of their students who have submitted a notification in My Slice Faculty Center.

## Learning Objectives:

### *(Numbers 1-6 refer to Bloom’s taxonomy levels[[1]](#footnote-1), e.g., as depicted in* [*https://upload.wikimedia.org/wikipedia/commons/2/24/Blooms\_rose.svg*](https://upload.wikimedia.org/wikipedia/commons/2/24/Blooms_rose.svg)*):*

### After taking this course, students will be able to:

1. Knowledge: Describe components of software specification and design documents
2. Comprehension: Distinguish between various kinds of diagrams and tools used for specification and design
3. Application: Apply unified methodology to specific problems of software specification and design
4. Analysis: Point out what’s missing in specification and design documents
5. Synthesis: Create software specifications; Design software architecture
6. Evaluation: Judge the completeness, consistency, and understandability of specification and design documents

**Detailed Course Objectives:**

|  |
| --- |
| Systems Development, Software Eng. Approaches, Structured Analysis & Design:   * 1: Describe various software engineering processes; Describe the structured analysis approach; Describe various Capability Maturity Model levels for software development organizations * 6: Distinguish principles behind different approaches, appreciating their relative advantages and disadvantages; Select S/E process depending on the nature of the application and the manner in which the software is likely be used and updated |
| Object-oriented systems analysis and design; the unified process & UML:   * 1: Describe the object-oriented systems analysis and design process; Describe the unified process and main components of UML * 2 : Review the advantages of the OO and Unified processes appreciating their relative advantages and disadvantages |
| Use-case diagrams & descriptions; Activity diagrams:   * 1: Describe the main components of use case diagrams, use case descriptions, and activity diagrams * 2: Distinguish between different kinds of entities and labels used in these diagrams and descriptions * 3: Apply unified process to specific software problems * 4 : Analyze high level problem descriptions and infer use-cases and activities to be described in a software specification * 5: Compose use-case diagrams, use-case descriptions, and activity diagrams * 6: Judge completeness and consistency of use-cases and activities described in an SRS |
| Structural modeling:   * 1: Describe the main components of structural models * 2: Distinguish between different kinds of classes; Interrelate classes * 3: Apply class description process to develop class-responsibility-collaboration (CRC) cards and class diagrams for specific software problems * 4: Analyze use-cases to determine relevant classes for a problem * 5: Compose CRC cards and class diagrams * 6: Judge completeness and consistency of class relationship descriptions |
| Behavioral modeling:   * 1: Describe the main components of behavioral models * 2: Distinguish interaction diagrams and behavioral state machines from structural diagrams * 3: Sketch sequence diagrams, communication diagrams, and behavioral state machines * 4: Analyze structural models to determine objects, operations, and messages * 5: Synthesize behavioral descriptions for new problems * 6: Judge completeness and consistency of behavioral descriptions, as well as their consistency with structural models |
| Elements of Design:   * 1: Identify the main components of software design * 2: Interrelate specifications and design documents; explain key concepts of coupling, cohesion, encapsulation, information hiding, inheritance, and polymorphism * 3: Apply object design activities; specify methods * 4: Verify and validate class and method designs * 5: Develop design documents * 6: Judge completeness and consistency of design documents, and their consistency with specification documents; assess design quality |
| Design Patterns; Class and Method Design:   * 1: Identify design patterns frequently used in software development * 2: Explain the usefulness of design patterns * 3: Apply design patterns to new problems * 5: Develop new designs that use one or more design patterns * 6: Compare previous designs with designs based on patterns |
| Interface design:   * 2: Explain key principles of designing both internal and external interfaces * 5: Develop internal and external interfaces * 6: Evaluate quality of interfaces |
| Data Management Design:   * 2: Explain differences between different data design strategies * 3: Map classes and objects to software data implementations * 4: Analyze efficiency, access speed, and size of data storage * 5: Design data stores * 6: Verify and validate data store designs |
| Project Management:   * 1: Describe the main components of a project management plan |
| Ethics in Software Engineering:   * 1: Describe the possible ways in which software development may be unethical. * 4: Analyze possible ethical problems with software instances. |

1. [*Bloom, B. S.*](https://en.wikipedia.org/wiki/Benjamin_Bloom); Engelhart, M. D.; Furst, E. J.; Hill, W. H.; [*Krathwohl, D. R.*](https://en.wikipedia.org/wiki/David_Krathwohl) (1956). Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. New York: David McKay Company. [↑](#footnote-ref-1)