CPSC 430 Software Engineering at the University of Mary Washington

CPSC 430 Software Engineering Syllabus

Professor: [Jennifer A. Polack,PHD](#) **Office**:Trinkle B21 **Office Hours:** 8-9am MW, 2-3pm M, 8 – 9:30am T and 11:30-12 R	**Term**: Fall 2016 **email**: JenniferPolack@gmail.com **Class**: TR 9:30 – 11:30
Description:	Writing a computer program is a challenging and creative experience, motivated by the desire to solve problems. The task of developing even a small computer program is not an easy one. Programmers are continually required to keep their attention focused on many different aspects of both problems and solutions. The software engineering course attempts to help students prepare for application development and their future jobs by teaching Object Oriented Analysis and Design, UML, and Rational Rose (CASE Tool). This class requires that students work in groups just as if they were in a company. Rarely do programmers design and program large-scale applications on their own.
Prerequisite:	CPSC 340 & 350

Text:	This course will use two required textbooks. Head First Software Development ISBN-13: 978-0-596-52735-8 UML Distilled ISBN-13: 978-0321193681 The UML Distilled text is available as an ebook and as a Kindle version. Course information, selected handouts, and all assignments are available electronically on Canvas.

Final Tuesday, December 13 Noon — 2:30 pm, we are using the 11 am time slot.

Course Goals and Objectives

At the end of the semester, you will be able to

- Identify and describe a specific problem requiring a software solution.
- Design an efficient system to solve the problem
- Use CASE tools to assist in designing and developing a
- Locate and learn to use useful
- Define a timeline for team members and project
- Present and communicate the current standings of a
- Maintain
- Develop a test

General Education Goals and Objectives

This course satisfies the Experiential Learning Requirement. The following goals are associated with this requirement.

- Students will be able to apply what was learned in coursework to new scenarios outside standard university courses
- Students will be able to identify their personal values

and learning goals and direct themselves by creating personalized learning experiences that may include alternative means of learning

- Students will be able to clarify and refine their understanding of their strengths and weaknesses in content of relevant disciplines and in skills such as time management, organization, professionalism, and so forth
- Students will be able to recognize their knowledge and lack of knowledge
- Students will be able to connect their undergraduate experiences and their post-graduation lives

Across the Curriculum Goals and Objectives

Writing Across The Curriculum

- to enhance students' understanding of course material by having them write frequently about that material and (b) to help students become better writers

Speaking Across the Curriculum

- Students will understand and be able to explain the conventions and expectations of oral communication as practiced within the discipline of the course
- Students will apply theories and strategies for crafting messages (verbal, nonverbal, and visual) for particular audiences and
- Students will be able to craft oral messages after a conscious process in which various options are reviewed and will be able to explain and support their
- Students will be able to metacommunicate about their own communication

How the Course Operates

This course will be run as an entrepreneurial software development company. As developers, you will work in teams of

2-4 on a substantial project. As the general manager, I am responsible for providing high level direction, obtaining and allocating resources, helping you identify and resolve problems, and evaluating your performance.

Throughout the course, we will consider at least three different viewpoints:

1. As a user or client, you want to know what the project will do, and whether it will be helpful to you. How could the system be easier to use or more flexible?
2. As a developer, you want to know how easy it will be to maintain or improve the system in the future. Consider factors such as modular design, documentation, etc.
3. As a manager or financial backer, you want to be able to monitor progress, decide whether the developers on the team are doing what they claim, and know whether the project will be finished on schedule.

Grading:

Exams 30%
Reports 30%
In Class Work (presentations, assignments, quizzes) 20 %
Final Project 20%

The goal is that you learn how to effectively develop software. You must have a large part of the software done and this will be demoed in the final presentation and graded appropriately, peer evaluation will also be factored into the final project grade. If the project is not complete then you will give a justification as to what is not completed and why it is not completed on the final turn in, which will be graded appropriately, this will be an automatic 10% off your final project grade if you have nothing to demo.

Letter grade distribution:

- 100-93%: A,

- 90-92%:   A-,
- 86-89%:   B+,
- 82-85%:   B,
- 79-81%:   B-,
- 76-78%:   C+,
- 71-75%:   C,
- 69-70%:   C-,
- 66-68%:   D+,
- 60-65%:   D,
- < 60%: F.

Mid Semester Grade:

The University provides the opportunity to provide grading feedback midway through the semester. This will take into account your midterm exam, homework, quizzes, and labs submitted up to that point. Any student receiving less than a 70% will receive an unsatisfactory (U) for their mid-semester grade. Students receiving a U should schedule a meeting with me to discuss how we may improve your performance in the class.

Disability Policy:

The Office of Disability Services has been designated by the University as the primary office to guide, counsel, and assist students with disabilities.

If you already receive services through the Office of Disability Services and require accommodations for this class, make an appointment with me as soon as possible to discuss your approved accommodations. Please bring your accommodation letter with you to the appointment.

If you have not contacted the Office of Disability Services and need accommodations, their phone number is 540-654-1266. The office is located in Lee Hall, Room 401.

CPSC 430 Honor Policy

Students are allowed to work together on all aspects of this class except the midterm and quizzes. However, for the some homework assignments, each student must submit his or her own write up/presentation, clearly stating the collaborators. Your submission must be your own. When in doubt, contact the instructors about whether a potential action would be considered plagiarism. If you discuss material with anyone besides the class staff, acknowledge your collaborators in your write-up. If you obtain a key insight with help (e.g., through library work or a friend), acknowledge your source and write up the summary on your own. **It is the student's responsibility to remove any possibility of someone else's work from being misconstrued as the student's**. Never misrepresent someone else's work as your own. It must be absolutely clear what material is your original work. Plagiarism and other anti-intellectual behavior will be dealt with severely. Note that facilitation of plagiarism (giving your work to someone else) is also considered to be plagiarism, and will carry the same repercussions.

Students are encouraged to use the Internet, literature, and other publicly available resources, **except the homework solutions and test (including quizzes, midterms, finals, and other exams) solutions, from past terms' versions of this course and other academic courses, whether at UMW and at other institutions**. To reiterate, the students are not allowed to view and use past homework and test solutions, unless explicitly distributed by the CPSC 430 staff as study material.

Whenever students use Internet, literature, and other publicly available resources, they must clearly reference the materials in their write ups, attributing proper credit. This cannot be emphasized enough: **attribute proper credit to your sources**. Failure to do so will result in a zero grade for the assignment and possibly a failing grade for the class, at the instructor's discretion. Copying directly from resources is

not permitted, unless the copying is clearly identified as a quote from a source. Most use of references should be written in the words of the student, placing the related work in proper context and describing the relevant comparison.

Students who violate University standards of academic integrity are subject to disciplinary sanctions, including failure in the course and suspension from the university. Since dishonesty in any form harms the individual, other students, and the university, policies on academic integrity have been and will be strictly enforced.

UMW Computer Science — Honor Policy

The following rules reflect the CPSC department's implementation of the UMW Honor Code. They were arrived at based on a strong census from the entire CPSC faculty.

Note that all of the rules below apply to all CPSC courses, unless explicitly overridden by the instructor in writing. ("in writing" means, for instance, in the class syllabus, or a Canvas announcement, or an email from the instructor).

1. Provide the honor pledge. All assignments, unless your instructor writes otherwise, must be accompanied by the honor pledge in some form. Your instructor will specify how to do this: typing it in code comments, writing it by hand and submitting it separately, entering it into Canvas when you submit your assignment, etc.
2. You must write all programs yourself (without help from others or from websites), unless specified. There are several team-oriented courses in our curriculum, and of course you will be allowed to work with other students on your team for those. Your instructor will explicitly identify those, however. In the absence of any specific instruction, you are *not* to communicate to others in any way about your assignments. You are also *not* to consult Google, StackOverflow, or any other website unless

permitted in writing.

3. Do not share your code with other students, either this semester, or in any future semester. Remember that *giving* unauthorized help violates the Honor Code just as much as *receiving* unauthorized help does.
4. Do not post your code or class materials anywhere. You may not upload your solutions to any publicly-available website, post part of your solution on StackOverflow or any similar site, or post assignments/notes/etc from the course, even if they were instructor-authored materials.
5. Explicitly cite any sources you use.

   - *If* your instructor states in writing that you are allowed to work with other students, you *must* indicate on your submission the names of your collaborators.
   - *If* your instructor allows you to use Google, StackOverflow, or any other electronic source, you *must* indicate on your submission the source and scope of the assistance you received. Individual instructors will give specific directions on how to do that; it might be creating a code comment that contains the URL and a synopsis of the code you used, for example.

6. Do not look at solutions from previous semesters. Professors evolve and reuse assignments over many years in order to perfect them. If someone does leave their code (or other materials) lying around from a previous offering of the course, you may *not* look at them when completing your own.
7. Be prepared to explain anything you submit. Your instructor may, at any time, call you in to his/her office to explain any part of your program. You will be expected to convincingly walk him/her through your code, demonstrating your thought process behind it. If you cannot, this may be considered an Honor Code violation.
8. When in doubt, ask your instructor what constitutes plagiarism. If you're not sure whether you need to cite

a source for a quotation in a paper, or list the URL of a website from which you got some code, *ask*. If you do not ask, and the instructor deems it to be unauthorized help, this may be considered an Honor Code violation.

9. Automated plagiarism detection tools may be used. Be forewarned that there are smart tools out there (like Stanford's "Moss," to name just one example) that can detect when two programs are so alike that rote copying must have occurred. These tools are clever enough not to be fooled by superficial changes, like adding white space or changing variable names. Be advised that instructors may (or may not) use such tools in order to identify illegal copying.

10. No assistance on exams/quizzes. Unless otherwise noted, all exams and quizzes must be taken *without* using any electronic device, hardcopy materials, etc.

Course Organization & Deliverables

List of topics covered & associated reading assignments

HF = Head First Software Development book

UML = UML Distilled book

Intro to software engineering   (HF Ch 1)

Systems Engineering   (HF Ch 1)

Software Process   (HF Ch 1)

Project Management   (HF Ch 1)

Software Requirements   (HF Ch 2 & 3 & 4)

UML Use Cases   (UML Ch 1, 2, and 9)

Uses Cases w/Rational Rose   (lab)

Software Design   (HF Ch 5)

Class Diagrams   (UML Ch 3 & 5)

Sequence Diagrams   (UML Ch 4)

Class Diagrams w/Rational Rose   (lab)

User Interface Design   (HF Ch 5)

Version Control   (HF Ch 6)

Implementation   (HF Ch 6 ½)

Testing   (HF Ch 7)

Verification & Validation   (HF Ch 7)

Test Driven Development   (HF Ch 8)

Ending Iteration/Next Iteration   (HF Ch 9 & 10)

Bug fixes & Real World SE   (HF Ch 11)

Group Project Deliverables:

- Turn in a team meeting report EVERY Monday. Print it out & bring it to class.
- Turn in each major document electronically

- Requirements Document
- Software Design Document
- Test Plan Document
- Testing Report (That will be turned in on the final project demo day)

- Turn in the final project in electronic format (USB drive)
- Make your group presentation/project demo during the last week of class and during finals

Individual Deliverables

- You must make at least two individual oral presentations in class over the course of the semester. You select the weeks that you want to present.
- There are several in class activities over the course of semester. Due dates will be given when lab is assigned.
- You will independently create a user interface and associated report. This will be due approximately mid-semester.

Course Project Overview:

In this course you will be applying what you learn to a semester-long group software engineering project. Playing the role of developers, you will work in teams of 2-4 on a substantial project. Your instructor will play the role of a high level manager and will be responsible for providing high level direction, obtaining and allocating resources, helping you to identify and resolve problems, and evaluating performance.

During the semester, we will consider at least three different viewpoints related to the group project:

1. The view of the client: A user or client asks questions like "what will the project do?" and "how will this be helpful to me?" Users consider how the system could be easier to use or more flexible.

1. The view of the developer: A developer will want to know how easy the software will be to maintain and how the system could be improved in the future. She or he will consider factors such as modular design, documentation, etc.

2. The view of the manager or financial backer: A manager or financial backer will want to be able to monitor progress, decide whether the developers on the team are doing what they claim, and assess whether the project

will be finished on schedule.

Team Management

Your instructor will assign you to teams of 2-4 persons, based in part on resumes and various preferences submitted during the first week of class. The teams will not be stagnant. You will work in a variety of different groups on a variety of different projects. In general, each team will have broad discretion regarding how to organize itself and manage its project; however the following ground rules apply:

1. Each member should have an equal share of both technical and non-technical responsibilities. Technical responsibilities include analysis, design, coding, technical documentation, and testing. **Each team member must participate in each of these technical activities.**
2. As a class we will create a code of conduct and a set of guidelines that specify minimum expectations for appropriate behavior for team members. If any member fails to meet these minimum expectations, the team should file a status update describing the problem and a plan to resolve it. If repeated problems with a member's actions occur, the team should request his or her termination by notifying the instructor.
3. Each team must meet at least once a week outside of class to discuss the current status of the project, assign tasks, and resolve any problems or conflicts. You must submit a meeting report to your instructor once a week on Tuesday unless otherwise requested by Dr Anewalt. The report should specify what tasks will be completed in the upcoming week or two, who is responsible for these tasks, and how they will be evaluated. See the example report on page 12.
4. Each member must periodically submit a confidential evaluation of all team members.

Project Deliverables:

You will need to turn in documentation at various stages of the project. Some examples of deliverables include:

Team Meeting Reports

- A team meeting report should be turned in every Monday. The report should specify what tasks will be completed in the upcoming week or two, who is responsible for these tasks, and how they will be evaluated.

Oral Status Reports:

- At the beginning of each class each team may be asked to provide a brief oral status report to the entire class. The topic for the status report will be announced at least a week prior to the scheduled presentation.

User Interface Prototype:

- Each individual group member must turn in a user interface prototype mid-semester. This is <u>NOT </u>a group effort.

Project Documentation

- This includes design documents, interface specifications, and comments in project code. The purpose of this documentation is to assist developers (ie. your team and others) in understanding how the product works. There will be 5 major writing assignments related to project documentation. Due dates for the writing assignments are indicated on the tentative course schedule.

Final Group Project Deliverable:

- At the end of the semester, your group must turn in a USB containing ALL of the following items:

- Report Folder: Containing updated versions of all reports including the project plan, requirements, design

and testing documents.

- Prototype Folder: Each person's prototype in a separate folder and the exe in the main folder
- Presentation folder: PowerPoint files for each presentation made by group members over the course of the semester
- Final code folder: All code
- A README file explaining how to install and run the program
- Project Explanation: A reflective paper describing the progress that was made toward completing the project. If the project was not completed, explain what work was not done and why. Explain how the project could be improved or expanded in the future.
- EXE file (if appropriate)

Team Reports Format

WEEKLY STATUS REPORT			
Customer Name:	ABC Inc.		
Project Name:	PROJECT Z	Project:	XYZ613
Project Phase:	Construction	Date:	97/04/24
Project Manager:	Jane Smith	Pages:	2
Distribution:	Gene Jones	From:	Jane Smith
Accomplished (this period):			

Process Assessment
* Completed changes to Process Assessment suites as a result of QA.
Help Text
* Loaded new Help Text into database.
* Included all Help Text calls in each module.
Staffing/Total Effort
* Incorporated changes as a result of QA.
Integration Testing
* Completed QA of pseudo IPG tables and TPA tables.
* Completed standardization of colours and hot keys and distributed to team.
* Completed Integration Testing, assignment of changes and updates to Integration Testing Log.
System Testing
* Completed set-up of System Test environment.
* Completed QA of Acceptance Test scripts.
Conversion
* Combined Project Index and IPG conversion modules and assigned to Scott Taylor.
Implementation
* Received feedback on first draft of Implementation Plan.
* Ordered AIX version 3.2.
* Finalized start of proof-of-concept of X-term software with Don Buckland, Paul Tait and Ray Wilander for week of May 4.
* Organized start of French translation by Guy Carpentier, Quebec City on April 28.
Other
* Adjusted new screens to cater for new non-proportional font.
* Continued with update of Help Text.
* Continued with Acceptance Test scripts.
* Completed rework of schedule and reassignment of tasks based on new direction in implementation.

Planned but not Accomplished (this period):

Integration Testing
* Complete changes arising out of Integration Testing.
Training
* Complete first draft of training materials.

Work Completed but not Planned (this period):

NA

Objectives (next period):

Integration Testing
* Complete changes arising out of
Integration Testing.
System Testing
* Start changes arising out of System
Testing.
* Start System Testing.
Help Text
* Determine cause of display problems in
help text.
Conversion
* Begin design for Project Index and IPG
conversion.
Other
* Complete first draft of Training
Materials.
* Continue with update of Help Text.
* Continue with Acceptance Test scripts.
* Finalize agenda for Steering Committee
Meeting on Monday May 4, 1997.

Problems/Warnings:

> 1.   Lack of a Final Architecture for Workstation is beginning to affect the design and development of (Import, Export, Generate Reports, Diagrams).
> 2.   Ordered AIX 3.2 and understand that INGRES 6.4/01/01 has been shipped to us.
> 3.   Require Highjack software to finalize Diagram Display.
> 4.   Generic error messages are included in the message table — however modules will pass specific messages as parameters to the message module to be included as variables in the message table. This will need to be addressed in preparing versions of TPA in other languages.

Name: _____

Grading:
Exams 30%, Reports 30%, In Class Work (presentations, assignments, quizzes) 20 %, Final Project 20%

The goal is that you learn how to effectively develop software. You must have a large part of the software done and this will be demoed in the final presentation and graded appropriately, peer evaluation will also be factored into the final project grade. If the project is not complete then you will give a justification as to what is not completed and why it is not completed on the final turn in, which will be graded appropriately, this will be an automatic 10% off your final project grade if you have nothing to demo.

Initial Each Line

_____I am aware that the grade I receive in this class is based on my exams, in class work, and all team activities (including reports, presentation, and the assigned software product).

_____I am also aware that having a completed working piece of software is not necessary to pass the class, but if nothing is done on the project and you have nothing to demo at the end you will be docked 10% off your final grade.

_____I am also aware that the real world project I am designing is graded based on the software lifecycle model, meaning I will actively participate in creating

- Requirements
- Specification
- Design
- Prototyping
- Implementation (As much as can be done in the time remaining in the semester)
- Testing (As much as can be done in the time remaining in the semester)

_____I am also aware that my client has agreed to volunteer their projects and realize that it may not get completed to the fullest potential that they want

_____I am aware that I leave class earlier when group work is suppose to be done that I will lose a point off my final grade each time I leave early. In addition, if I do not show up with a documented reason I will lose a point off my final grade each time I am not in class when group work in being done and assigned. Class time is given to work on the project, you do not have the option of leaving the work must be done during that period. If everyone continues to leave early without completing the group work no class time will be given to work on the project.

_____Finally I will follow the process guidelines set forth in class and that I have a partial working product at the end of the semester but know that if the software is not then I will write a justification as to why it was not completed and will be graded on that justification as part of

the final USB grade.

Signature:_____
____