

CS 411W

Professional Workforce Development II

Writing Intensive • Team-Based Software Development • Professional Communication • Fall 2026

Course	CS 411W Professional Workforce Development II
Semester	Fall 2026
Course Format	Scheduled meetings via Zoom; course materials and activities in Canvas
Meeting Access	Canvas → Course Collaboration Tool
Recordings	Canvas → Media Gallery
Instructor	Dr. Sumaya Sanober
Email	ssanober@odu.edu
Office	Monarch Hall 2123D
Zoom Office Hours	book here, Sumaya Sanober, ODU CS Dept. Twice a week, on Tuesday (11:00 AM-12:00 PM) and Thursday (12:00 PM-1:00 PM)
In-Person Office Hours	Wednesday, 12:00 PM–1:00 PM

Welcome to CS 411W: This course continues the project you began in CS 410 and moves your team from design into detailed development, testing, documentation, and professional demonstration. Success depends on steady individual contribution, clear communication, and sustained teamwork throughout the semester.

1. Basic Course Information

1.1 Catalog Description

Laboratory work required. Students write professional and non-technical documents and continue the development of the project defined in CS 410. Written work is reviewed and returned for corrective rewriting. Students will design and develop a project prototype, demonstrate the prototype to a formal panel, deliver formal product specifications, and prepare a draft formal grant proposal. This is a writing-intensive course.

1.2 Overall Description

The Professional Workforce Development course sequence (CS 410 and CS 411W) expands upon the experimental and design approach of typical computer science curricula by addressing the creativity and productivity required for business and industrial applications. Students in this course sequence engage in projects that investigate each stage of transforming a creative idea into an innovative product.

Writing can comprise a substantial portion of a computer scientist's professional activities. The evolution of software engineering into a team-centric process—often across geographic distances—requires the sharing of a wide variety of documentation with attention to version control and version management.

This course is the writing-intensive (W) course for the major, which requires 51% or more of the computed grade to be based upon writing assignments. You must demonstrate the ability to write clearly and accurately for a technically oriented audience. You will complete both graded and ungraded writing exercises.

You will write professional and technical documents and continue development of the project defined in CS 410. Written work will be reviewed and returned for corrective rewriting. Students will design, develop, test, document, and formally demonstrate a project prototype.

1.3 Prerequisites

Students must have earned a grade of C or better in one of the following:

- ENGL 211C
- ENGL 221C
- ENGL 231C

Students must also have earned a grade of C or better in all three of the following:

- CS 330
- CS 350
- CS 410

1.3.1 General Programming Knowledge

Students should be familiar with basic programming techniques that are largely independent of any specific programming language:

- Using editors, compilers, and other basic software-development tools.
- Basic software design, including stepwise refinement and top-down design.
- Software testing, including scaffolding code (stubs and drivers), selection of test cases for black-box testing, and head-to-head testing.
- Debugging, including debugging output, automatic debuggers, breakpoints, execution tracing, and reasoning backward from a failure location to the faulty code responsible for the failure.

1.3.2 C++ & Java Knowledge

I will assume that you are familiar with the basics of C++ or Java. This includes:

- Statements and control-flow constructs.
- Built-in data types.
- Arrays, pointers, pointers to arrays, and linked lists.
- The use and writing of functions.
- Basic use of structs and/or classes for implementing abstract data types.

1.3.3 Python 3 & Rust Knowledge

Prior knowledge of Python and Rust is neither expected nor assumed. In general, computer science students at the 400 level should be able to pick up new programming languages with moderate effort.

Useful resources: [Python 3 documentation](#) • [Rust learning resources](#) • [ODU CS 263 materials](#)

1.3.4 Unix/Linux

All students in the course will receive accounts on the CS Department network, and knowledge of how to work with the Linux servers is part of the course prerequisites. This course does not require familiarity with shell scripting. All other topics from CS 252 are required.

Some assignments will require use of software available only on the Linux servers. Others may require—or be simplified by—the use of the X Window System.

1.3.5 General Computer Literacy

You will study techniques for preparing professional-quality software documentation. The key embedded word in “software documentation” is “document.” Students taking this course should be able to use word processors and other common tools to produce high-quality documents, including mixing text and graphics in a natural and professional manner.

1.4 When and Where

Scheduled meetings will be held via Zoom. Meeting information can be found in Canvas under the Course Collaboration Tool. Recordings of these meetings will be available under Media Gallery.

1.5 Instructor

Dr. Sumaya Sanobar is an Assistant Professor in Teaching in the Department of Computer Science at Old Dominion University (ODU). She holds a Ph.D. in Computer Science, with specialization in Artificial Neural Networks, and has more than 15 years of experience in teaching and higher education. Her teaching experience spans a wide range of computer science areas, including Artificial Intelligence, Machine Learning, Software Development, Professional Workforce Development, computational problem solving, and student-centered computing education.

Dr. Sanobar is also an interdisciplinary researcher whose work brings together Artificial Intelligence, Machine Learning, data-driven decision-making, and their applications to real-world societal challenges. Her current research interests include AI for healthcare and public health, climate and flood-risk analytics, trustworthy human-AI collaboration, neural networks, and the use of intelligent systems to support evidence-based decision making. She has contributed to interdisciplinary research initiatives involving computer science, public health, environmental resilience, and community-centered technology.

In addition to teaching and research, Dr. Sanobar is actively involved in curriculum development, student mentoring, undergraduate research, and initiatives that prepare students for academic and professional success. She is particularly interested in making computing education accessible, engaging, and career focused while helping students develop the technical, communication, research, and problem-solving skills needed in today's rapidly evolving technology landscape.

1.5.1 Contact Policy

All email related to this course should include "CS 411W" in the subject line. This helps flag your message for faster attention.

Email is my preferred mode of communication with students. Do not hesitate to email your questions; I am happy to provide guidance when needed. I will respond to emails within 24 hours on weekdays and within 48 hours on weekends and holidays. For course-related communication, please use your ODU email account.

1.5.2 Office Hours

For Zoom office hours on Tuesday and Thursday, book an appointment through the [Sumaya Sanobar, ODU CS Department booking page](#). In-person office hours are Wednesday from 12:00 PM to 1:00 PM in Monarch Hall 2123D.

2. Required Materials

2.1 Textbooks

There is no required textbook. All readings will be provided through Canvas and the course site.

2.2 Supplemental Materials

All supplemental resources and materials will be available in Canvas.

2.3 Technology Requirements

2.3.1 Computer Accounts

Students will need an account on the CS Department Linux network to participate in this class. This account is separate from any University-wide account you may have through ODU Information Technology Services (ITS).

If you have had a CS Unix account in the recent past, it should still be active with your login name, password, and files unchanged. If you have forgotten your password or believe your password is no longer active, use the CS Account Management password-reset service.

Account resources: [CS Account Management](#) • [Create a Unix Account](#)

If your account cannot be restored after using the available self-service options, contact the CS Department systems staff at root@cs.odu.edu. All students are responsible for ensuring that they have a working CS Unix account prior to the first assignment.

3. Course Objectives

At the end of this course, students will be able to:

1. Recognize the value of collaboration in the field of computer science.
2. Plan and execute the development of a major software project.
3. Identify the order in which tasks should be completed either in parallel or in sequence when completing large projects.
4. Establish communication strategies to stay on task, complete team-based deliverables, and conduct effective team meetings asynchronously.
5. Establish effective asynchronous and synchronous communications with faculty and industry mentors.
6. Prepare collaborative outlines of the required written assignments.
7. Prepare quality written papers based upon collaborative outlines.
8. Document individual progress, challenges, concerns, and successes.
9. Recognize the need for the four types of professional writing addressed in the course as professionals and academics.
10. Utilize the assignments to develop high-quality group projects.
11. Recognize the value of prototyping in the development of software projects.

4. Course Schedule

The detailed course schedule is available in Canvas under Modules and Calendar. A consolidated Fall 2026 course summary is included in Section 8 of this syllabus. Canvas remains the authoritative source for current assignment instructions, links, and any approved schedule updates.

5. Grading

The following components and weightings are used for course projects and activities:

Graded Element	Description	Weight	Individual / Group
Writing & Discussion Exercises	Discussion activities	5%	Individual
Lab 1	Descriptive/Persuasive Paper	15%	Individual
Lab 2	Prototype Product Specification	15%	Individual / Group
Lab 3 & Lab 4	Prototype Test Plan & User Manual	15%	Individual / Group
Prototype Setup & Evaluations	Product Laboratory Prototype	5%	Individual / Group
Prototype Development	Product Laboratory Prototype	35%	Individual / Group
Website	Product and Course Documents	5%	Group
Project Poster	Project Synopsis	5%	Group

Grading Schema

Percent	Letter Grade	4-Point Value
≥ 94	A	4.0
90–93	A–	3.7

Percent	Letter Grade	4-Point Value
87–89	B+	3.3
84–86	B	3.0
80–83	B–	2.7
77–79	C+	2.3
74–76	C	2.0
70–73	C–	1.7
67–69	D+	1.3
64–66	D	1.0
60–63	D–	0.7
< 60	F	0.0

5.1 Presentations and Discussion Activities

Throughout the semester, several discussion and presentation activities will be organized to ensure continuous engagement and progress on your capstone projects. These will include periodic informal presentations where you will showcase development progress. These sessions provide opportunities for feedback, help you refine your project incrementally, and prepare you for the formal prototype presentation at the end of the course.

Required discussion-board activities will also be assigned. Details for both presentations and discussions will be provided through Canvas Announcements and discussed in course meetings. These activities serve a dual purpose: they allow for evaluation of each team's project as a whole and assess each student's understanding of and contribution to the project. Participation in these discussions and presentations is important for evaluating individual contribution.

The semester will conclude with a formal presentation of your Final Prototype, attended by external evaluators and mentors. This final assessment will showcase both the development effort reflected in your prototype and your ability to communicate your work effectively to an audience of professionals.

6. Course Policies

6.1 Attendance

Meeting attendance is not required. If you are unable to attend a scheduled meeting, it is your responsibility to:

1. Watch the recording of any meeting that you are unable to attend.
2. Follow up with your team regarding any team feedback provided during recitation or project meetings.
3. Contribute meaningfully to the development of presentation, written, and prototype materials.

Students must meet all course deadlines and be present for required formal and informal Prototype Demonstrations.

6.2 Due Dates and Late Submissions

Late papers, assignments, prototype contributions, and presentations/demonstrations will not normally be permitted. Exceptions will be made only in unusual and unforeseeable circumstances beyond the student's control. When a conflict is foreseeable, arrangements must be made before the due date.

6.3 Group Membership (From CS 410 to CS 411W)

In CS 410, you were assigned to a team. Your team developed an idea and designed a prototype. Barring specific conditions, you will remain with your CS 410 team.

You will be assigned to a new team if:

- It has been more than one semester since you completed CS 410.
- You are retaking CS 411W.
- You are completing CS 411W during the summer.
- Your original CS 410 team is no longer complete because of a reduced number of members or because it does not have enough members to meet course requirements.

You may be assigned to a new team, at the instructor's discretion, if you are unable to work with your CS 410 team. Note the use of "may."

6.4 Group Contribution

You will work as part of a team for most of the semester (generally from Week 2 onward). Your grade will include and be based on:

1. Active participation within your team.
2. Active contribution to your team.
3. Active engagement with your team.

Procrastination and just-before-the-deadline submissions are detrimental to team dynamics and may result in lowered grades.

6.5 Prototype Contribution

The Prototype (Software Solution) is a core deliverable of CS 411W. All students must contribute to development by:

1. Contributing to the group's efforts to learn selected tools and technologies.
2. Contributing code continuously throughout the semester.
3. Contributing meaningful code.
4. Communicating with the group throughout the semester.

Critical requirement: Any student who fails to contribute effectively to the Prototype will not pass the course.

6.6 Civility Among Team Members

You will be working with your team for many weeks, and substantial communication is expected among team members.

In accordance with the [Monarch Creed](#) and the [Code of Ethics](#), all students are expected to maintain civility in their dealings with one another.

Language that is abusive, harassing, or threatening to members of the class, or that fosters high levels of personal and emotional anxiety, may—at the instructor's discretion—result in expulsion from the team. Given the importance of the team project to this course, that is likely to result in a failing grade. Egregious or repeated violations will be referred to appropriate authorities for possible disciplinary action.

6.7 Unsatisfactory Performance

Your team will draft a Team Contract at the start of the semester. A team may petition to remove a member for failure to live up to the Team Contract, following the process below.

6.7.1 Formal Warning

To issue a formal warning:

1. A majority of the team members who would remain, should the team member eventually be removed, must approve the warning.

2. A petition to warn the team member must be sent to the instructor and approved by the instructor. The petition must document the reasons for the warning, including the Team Contract expectations that have been violated and the work the person has failed to complete. The petition need not be lengthy, but it must document the reasons. The instructor will share these reasons with the person being warned.
3. Upon receipt of the petition, the instructor will decide whether to issue the warning and will notify all persons involved.

6.7.2 Petition to Fire a Team Member

To fire a team member:

1. The team must have issued a formal warning to the team member at least one week before.
2. A majority of the team members who would remain must approve the firing.
3. A petition to fire the team member must be sent to the instructor and approved by the instructor. The petition must document the reasons for the firing, including Team Contract expectations that have been violated and work the person has failed to complete. The instructor will share these reasons with the person being considered for removal.
4. Upon receipt of the petition, the instructor will decide whether to remove the team member and the date on which the removal becomes effective.

6.7.3 Firing Aftermath

If a team member is removed, the team may remove that student's name from the access list to the team repository, tracking board, and shared documents. If the removed team member is the owner of a repository or board and therefore cannot be removed, the team is responsible for making a fork or copy of the material in order to move forward.

The removed team member will retain their own copy of the team repository and their knowledge of what was in any shared documents. Any disputes over access to formerly shared materials will be resolved by the instructor.

Due to the group nature of the course, a removed student will receive zeroes for any remaining group-based grades, including Prototype Development.

6.8 Academic Honesty

Everything turned in for grading in this course must be your own work. If an assignment is explicitly described as a team assignment, it must be the work of the authorized team members only.

The instructor reserves the right to question a student orally or in writing and to use the student's demonstrated understanding of an assignment and submitted solution as evidence when evaluating potential academic-integrity violations. Violations will be reported to the Office of Student Conduct & Academic Integrity for consideration of possible action.

Students who contribute to violations by sharing their code or designs with others may be subject to the same penalties. Students are expected to use standard Unix protection mechanisms (for example, `chmod`) to keep individual assignments from being read by classmates. Failure to do so may result in grade penalties.

This policy is not intended to prevent students from providing legitimate assistance to one another. Students are encouraged to help one another learn to use the operating system, understand programming-language concepts, and work through general issues related to course subject matter. Students should avoid explicit discussion of solutions to a particular graded programming assignment and, under no circumstances, should students show one another code for an ongoing individual assignment or discuss such code in detail.

Policy on the Use of AI Tools in Course Assignments

1. General Usage of AI Tools

Students are permitted to use AI tools such as ChatGPT for enhancing productivity in coding and writing tasks. However, the fundamental expectation is that all submitted work—whether code or written text—must predominantly reflect the student's own understanding and originality.

2. Specific Guidelines for Coding

If AI tools are used to generate or adapt code, students must cite the specific tool and any external code sources directly within code comments, a file header, or the pull-request description, as appropriate.

Example of citing code generated with AI assistance:

```
// Retry logic generated with the assistance of GitHub Copilot,  
// reviewed and modified to handle the timeout case.  
int attempts = 0;  
while (attempts < MAX_RETRIES) {  
    try {  
        return client.fetch(endpoint);  
    } catch (TimeoutException e) {  
        attempts++;  
        Thread.sleep(BACKOFF_MS * attempts);  
    }  
}
```

For larger generated components, a citation at the top of the file or in the pull-request description is acceptable. You are responsible for every line you commit under your name. You must be able to explain how the code works, why it is structured the way it is, what it depends on, and what would break if it changed. If you cannot explain code you submitted, it does not count as your contribution, regardless of whether it runs. You may be asked to walk through any part of your codebase during a check-in or demonstration.

3. Specific Guidelines for Writing

The use of AI tools such as ChatGPT in writing assignments is allowed for tasks such as drafting, revising, and style enhancement. However, all AI-generated material must be critically reviewed, edited, and integrated by the student so that the final work reflects personal insight and academic rigor.

Students must retain a precise copy of the prompt material supplied to the AI tool and be prepared to provide that prompt upon request by the instructor. Any direct use of AI-generated text must be cited appropriately. A simple disclosure such as “This text was generated with the assistance of ChatGPT” may be used when appropriate to the assignment and citation requirements.

4. Integrity and Originality

Direct copying of AI-generated content without critical engagement or significant personal input will be scrutinized. Submissions that lack original thought, or that rely heavily on AI without substantial student input, will not be accepted. For example, submitting an unedited AI-generated Lab 1 report—even if cited—does not demonstrate the student’s own work and understanding.

Current AI tools can generate entire features, not just snippets. The requirement is the same as for writing: the work must reflect your understanding. For code, that understanding must be demonstrated by your ability to explain it.

5. Prohibitions

Posting details of course assignments, projects, or tests on online forums, bulletin boards, homework sites, or similar platforms in order to solicit help is strictly prohibited.

All work turned in for grading must be demonstrably your own. Failure to meet these standards, even with proper citation, may result in a grade of zero for the assignment if the instructor determines that the work does not sufficiently demonstrate the student’s knowledge and effort. The governing principle remains: “Everything turned in for grading in this course must be your own work.”

7. University Policies

7.1 Code of Student Conduct and Academic Integrity

The Office of Student Conduct & Academic Integrity (OSCAI) oversees administration of the student conduct system, as outlined in the Code of Student Conduct. Old Dominion University is committed to fostering an environment that is safe and secure, inclusive, and conducive to academic integrity, student engagement, and student success. The

University expects students and student organizations/groups to uphold and abide by standards embodied in core values that include personal and academic integrity, fairness, respect, community, and responsibility.

University resource: [Code of Student Conduct](#)

7.2 Honor Pledge

By attending Old Dominion University, you have accepted the responsibility to abide by the Honor Pledge:

ODU Honor Pledge: I pledge to support the Honor System of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member of the academic community it is my responsibility to turn in all suspected violations of the Honor Code. I will report to a hearing if summoned.

7.3 University Email Policy

Email is considered an official means for University communication. Old Dominion University provides each student an email account for teaching and learning, research, administration, and service. Students are responsible for maintaining and regularly checking their official ODU email account.

Policy reference: [University Computing Standards — Student Email](#)

7.3.1 Student Email

Students should be aware of expectations associated with official University email use. The ODU email account is an official point of contact for University correspondence.

7.3.2 Email Account Activation

It is the responsibility of every eligible student to activate MIDAS, the Monarch Identification and Authorization System, in order to obtain email access.

7.3.3 Expectations Regarding Use of Email

Students are expected to check their official email account frequently and consistently in order to stay current with University communications. Mail sent to the ODU email address may include notifications of University-related actions, including academic, financial, and disciplinary actions.

Additional information: [ODU Student Computing](#)

7.3.4 Educational Uses of Email

University offices and instructors cannot validate that an email communication is from an ODU student unless it comes from a valid ODU email address. If students send mail from non-ODU accounts, faculty and staff are not obligated to respond and may request that official ODU email accounts be used.

7.4 Withdrawal

Enrollment in this course indicates your acceptance of its teaching focus, requirements, and policies. Review the syllabus and course requirements as soon as possible. If you believe that the nature of this course does not meet your interests, needs, or expectations; if you are not prepared for the amount of work involved; or if you anticipate that assignment deadlines or adherence to course policies will constitute an unacceptable hardship, you should consider dropping the course by the applicable drop/add deadline.

University resources: [ODU Academic Calendar / Schedule Information](#) • [Office of the University Registrar](#)

7.5 Educational Accessibility

Old Dominion University is committed to ensuring equal access to all qualified students with disabilities in accordance with the Americans with Disabilities Act. The Office of Educational Accessibility (OEA) is the campus office that works with students who have disabilities to provide and/or arrange reasonable accommodations.

If you experience a disability that will impact your ability to access any aspect of this class, please present an accommodation letter from OEA so that we can work together to ensure that appropriate accommodations are available to you.

If you anticipate barriers to learning and/or testing in this class but do not have an accommodation letter, please consider scheduling an appointment with OEA to determine whether academic accommodations are appropriate.

Office of Educational Accessibility: 1021 Student Success Center • (757) 683-4655

8. Fall 2026 Course Summary

The schedule below consolidates the dates provided in Canvas. Unless otherwise noted, listed assignments are due by 11:59 PM. Canvas remains the authoritative source for detailed instructions, submission links, and approved schedule changes.

Date	Course Activity / Milestone	Due
Sat Aug 22	Start Module 1 — Online Orientation and Course Introductions	11:59 PM
Mon Aug 24	Classes Start	—
Tue Aug 25	Double Check Registration	11:59 PM
Tue Aug 25	Start Module 2 — Prototype Foundations	11:59 PM
Wed Aug 26	Personal Introduction Post	11:59 PM
Sun Aug 30	Create GitHub Repository	11:59 PM
Sun Aug 30	Group Contact & Contract	11:59 PM
Sun Aug 30	Lab 1 Draft — Set Up Formatting & Structure	11:59 PM
Mon Aug 31	Lab 1 Outline Updates & Review	11:59 PM
Tue Sep 1	Start Preparations for Prototype Demo 1 (Informal Discussion)	11:59 PM
Sun Sep 6	Initial Website Update	11:59 PM
Sun Sep 6	Lab 1 Draft — Update Section 1	11:59 PM
Sun Sep 6	Prototype Demo 1 — Date Selection	11:59 PM
Sun Sep 6	Story & Issue Tracker Selection	11:59 PM
Mon Sep 7	Labor Day — No Classes	—
Sun Sep 13	Lab 1 Draft — Update & Add Section 2	11:59 PM
Sun Sep 20	Lab 1 Draft — Update & Add Section 3	11:59 PM
Mon Sep 21	Start Module 3 — Detailed Design	11:59 PM
Thu Sep 24	Start Preparations for Prototype Demo 2 (Informal Discussion)	11:59 PM
Sun Sep 27	Lab 1 — Version 1	11:59 PM
Sun Sep 27	Self Evaluation — Lab 1	11:59 PM
Tue Sep 29	Interim Peer Evaluation 1	11:59 PM
Tue Sep 29	Prototype Demo 1 — First Informal Presentation	11:59 PM
Tue Sep 29	Moving Forward from Demo 1 to Start Preparation for Demo 2 (Informal Discussion)	11:59 PM
Fri Oct 2	Website Update — Lab 1	11:59 PM

Date	Course Activity / Milestone	Due
Sun Oct 4	Prototype Demo 2 — Date Selection	11:59 PM
Fri Oct 9	Self Evaluation — Prototype Midway	11:59 PM
Sat Oct 10	Fall Break — No Classes	—
Sun Oct 11	Fall Break — No Classes	—
Mon Oct 12	Fall Break — No Classes	—
Tue Oct 13	Fall Break — No Classes	—
Sun Oct 18	Lab 2 — Sections 1 & 2 (Individual)	11:59 PM
Thu Oct 22	Prototype Demo 3 — Date Selection	11:59 PM
Thu Oct 22	Start Preparations for Prototype Demo 3 (Informal Discussion)	11:59 PM
Fri Oct 23	Draft Poster	11:59 PM
Sun Oct 25	Lab 2 — Section 3 (Group)	11:59 PM
Mon Oct 26	Start Module 4 — Testing	11:59 PM
Tue Oct 27	Interim Peer Evaluation 2	11:59 PM
Tue Oct 27	Prototype Demo 2 — Second Informal Presentation	11:59 PM
Fri Oct 30	Revised Poster	11:59 PM
Sun Nov 1	Lab 2 — Sections 1 & 2 (Individual, Version 2)	11:59 PM
Mon Nov 2	Prototype Demo 3 (Discussion)	11:59 PM
Tue Nov 3	Election Day — No Classes	—
Wed Nov 4	Lab 3	11:59 PM
Fri Nov 6	Website Update — Lab 2 & User Manual	11:59 PM
Sun Nov 8	Lab 2 — Section 3, Version 2 (Group)	11:59 PM
Sun Nov 8	Self Evaluation — Lab 2	11:59 PM
Mon Nov 9	Start Module 5 — Customer Interaction	11:59 PM
Fri Nov 13	Interim Peer Evaluation 3	11:59 PM
Mon Nov 16	GitHub Repository URL	11:59 PM
Wed Nov 18	Lab 4	11:59 PM
Thu Nov 19	Formal Prototype Demo — Date Selection	11:59 PM
Fri Nov 20	Self Evaluation — Lab 4	11:59 PM
Mon Nov 23	Start Module 6 — Lessons Learned	11:59 PM
Wed Nov 25	Thanksgiving Break — No Classes	—
Thu Nov 26	Thanksgiving Break — No Classes	—
Fri Nov 27	Thanksgiving Break — No Classes	—
Sat Nov 28	Thanksgiving Break — No Classes	—
Sun Nov 29	Thanksgiving Break — No Classes	—
Thu Dec 3	Website — Final Update	11:59 PM

Date	Course Activity / Milestone	Due
Fri Dec 4	Final Peer Evaluation	11:59 PM
Mon Dec 7	Prototype Follow Up	11:59 PM
Mon Dec 7	Self Evaluation — Prototype End of the Semester	11:59 PM
Fri Dec 11	Last Day of Classes	—

Student Acknowledgement

I have read this syllabus and understand the course expectations, requirements, grading structure, communication practices, academic-integrity standards, and team responsibilities described in this document.

Student Name	Date
Signature	ODU Email
Team Name (if assigned)	Section / CRN