

**University of Wisconsin Oshkosh
Computer Science Department**

**CS 221: Object-Oriented Design & Programming
Spring 2009**

----- Course Syllabus -----

COURSE SCHEDULE

		Days	Time	Location
Section 0081	Online Chat *	T or TH	6-7:00pm	UWO D2L Online Chat
	UWO Open Lab	SAT Feb 13	10am-12noon	Halsey Science 101C
		SAT Mar 7	10am-12noon	
		SAT Apr 4	10am-12noon	
		SUN Apr 19	1-3pm	
		SAT May 9	10am-12noon	

**Day/time to be scheduled during first week of classes*

INSTRUCTOR Dr. Kathy (Kate) S. Faggiani
Computer Science Department

EMAIL faggianik@uwosh.edu

PHONE (920) 424-2069

IN-PERSON OFFICE HOURS*(HS 218):

M 11:25am-1:45pm

TTH 12:00-1:20pm

Or use Google CHAT: faggianik@gmail.com whenever I'm online.

*Also available other times by appointment

PREREQ A grade of C or better in one of the following: MATH 104, MATH 108, MATH 206, or CS 142; or qualifying for MATH 171 via the Mathematics Placement Exam.

COURSE DESCRIPTION

A first course in problem solving, software design, and computer programming, using the Java language. Problem solving/software design techniques will be drawn from: flow charts, pseudo code, structure charts, and class diagrams. Data structures and algorithms include arrays, character strings, searching, and sorting. Programming topics include: data types, project statements, standard input/output, selection, repetition, functions/methods, parameters, scope of identifiers, data file input/output, recursion, and simple GUIs. (Source: University of Wisconsin Oshkosh Undergraduate Bulletin, 2007-2009, p. 128.)

PURPOSE OF THE COURSE

This course is a required course in the Computer Science, Mathematics (Applied emphasis), and Management Information Systems programs. It also appears as an elective course in several other degree programs. It introduces object-oriented design and programming concepts widely used in problem-solving with computer programs and in software engineering. The course emphasizes the

core concepts in computer programming that support problem-solving, and provides a foundation for further study in the field of Computer Science and related disciplines.

COURSE OVERVIEW

Object-oriented design and programming, like any computer programming language or in-depth learning experience, requires active involvement by the learner. This section of the course is an online learning experience that maximizes learner involvement. Course information is posted in D2L, accessible from: <https://uwosh.courses.wisconsin.edu/>. Note that username/password are the same as UWO Titanmail. Click the Titanmail link for information on your username/password. Students are expected to review and follow the course instructions posted in D2L on a bi-weekly basis. Course activities will involve programming, quizzes to assess learning of terminology and key concepts, online discussions, online chats, and optional open labs held face-to-face (FTF) on the UWO campus in the Halsey Science building computer lab, HS101C. The online chats will be offered at two times during the week – 6-7pm on Tues or Thurs. Participants must attend ONE of the two chats, and may attend both if desired.

LETTER TO THE STUDENT

Welcome to Object-Oriented Design and Programming !! I hope you're looking forward to an interesting and FUN semester. I've added this letter to the syllabus to share some additional perspectives on the course that you may find informative and useful.

While this course focuses on object-oriented design and programming skills, the course builds a valuable foundation of knowledge for further computer programming, computer science, and management information systems courses. Good programming skills are based on a systematic and organized approach to thinking about problems and their solutions. This thought process is referred to as "computational thinking". We will use an object-oriented approach to analyze and solve problems, and we will create solutions to problems using the Java computer programming language.

Although many different programming languages and tools can be used to illustrate computational thinking, object-oriented design, and programming skills, Java is exceptionally well-suited to the task. It is also a language that has grown in popularity and use in both scientific and business computer software development. As of August 2008, there were over 15,000 employment opportunities nationwide) for people who have varying degrees of proficiency with the Java programming language, and over 150 of these jobs were in Wisconsin (<http://www.dice.com> search with keyword Java retrieved on Monday, August 25, 2008). The U.S. Bureau of Labor Statistics indicates that computer software engineers, specifically those engaged in application development, are among the top 20 occupations with the highest projected employment increases during 2006-2016 (<http://www.bls.gov/oco/images/ocotic08.gif>). Average salaries for entry level programmer analysts, software developers, and software engineers vary widely depending on industry and geographical location, but a starting salary in the mid- to upper-40s is fairly common in our area. Salary information for a variety of computer-related occupations can be found in the ComputerWorld 2007 Salary Survey at: http://www.computerworld.com/spring/salary-survey.htm?activeYear=2007&type=job_level¶meter=0 .

In addition to providing potential employment opportunities, developing strong computational thinking skills helps foster critical thinking and problem-solving skills that are useful in a variety of different occupations, continuing undergraduate study, or graduate school study. Please note that the development of these skills is more challenging than simply acquiring knowledge of specialized vocabulary and memorizing facts. Analysis, problem-solving, and computer programming are activities in which you must actively engage, thus a major part of the learning experience will require you to spend time programming with Java and NetBeans on your computer or in a campus computer lab. Please plan accordingly to devote time to your learning process in this course. One rule of thumb applied to most university level courses is 3 hours per week outside of class for EACH CREDIT of a course in which you are enrolled. Nine hours per week PLUS three hours of class/lab time is a reasonable **minimum** expectation of time you'll need to devote to any university level course. You can expect to devote 9-12 hours per week to this course, and to expect this course to be much more challenging than other "introductory" level courses.

I hope this brief letter provides you with additional information about what the course involves, the practical value of learning to program with Java and object-oriented techniques, and the degree of challenge the course will provide for most learners; however, I also want you to know that object-oriented design and programming with Java is a lot of FUN (and challenging)! I try not to do anything that's not fun, and I hope you'll find that this course is no exception to the rule! Of course, it can be frustrating learning to program (those of you who have already had a shot at it know exactly what I mean), but hopefully the fun and frustration will balance out and you'll learn a great deal in the process!

Also, if you're wondering what to call me since my last name can be a real mouthful, my previous students have called me by my first name/nickname Kate, Dr. F, Dr. Kate, or those with a bit of Italian in their background have used the full-on "Dr. Faggiani". I try not to be too formal, so feel free to use my first name or whatever you find most comfortable.

I look forward to working with you this semester!

Your Instructor, Java Tour Guide, and "Personal Assistant" for CS 221,
Dr. Kathy (Kate) Faggiani

WHAT YOU SHOULD KNOW BEFORE STARTING THIS COURSE

This is a first course in computer programming, but shouldn't be your first experience using a computer! In addition to completing the course prerequisites, you are expected to start this course knowing how to do the following:

1. Use a web browser to go to a specific location on the internet.
2. Open and print documents and web pages from the internet.
3. Download and install software from the internet (following directions provided by the supplier).
4. Start and shutdown an application.
5. Create/open/save documents using MS Word.

6. Move, copy, rename, compress/zip/archive and decompress/unzip/unarchive files.
7. Send and receive messages in UWO Titan Mail, including downloading and uploading attachments.
8. Login and navigate in D2L.
9. Upload files to a D2L digital dropbox.
10. Check your grades in D2L.

Please contact the instructor if you have questions about how to perform these tasks.

MATERIALS AND RESOURCES

- REQUIRED TEXTBOOK** Savitch, W. & Carrano, F. (2009). Java: An Introduction to Problem-Solving & Programming, Fifth Edition. ISBN: 978-0-13-607214-0. This book is available at the UWO bookstore and also as an eTextbook at almost half the cost of hardcopy. DO purchase a NEW book so that you get access to the online VideoNotes. You can purchase the eTextbook subscription online at: <http://www.coursesmart.com/978013609316>
- SOFTWARE** NetBeans IDE 6.5 (also in campus computer labs). Download free for your own computer from: <http://java.sun.com/javase/downloads/netbeans.html>. Select the **Java SE Development Kit 6u11 and NetBeans IDE 6.5 Bundle** to download both the NetBeans development environment and JDK.
- OTHER** Online access to VideoNotes and GOAL.

LEARNING OBJECTIVES AND ASSESSMENT OF STUDENT LEARNING

Learning objectives are statements of what the student will be know and/or be able to do following successful completion of the course . The learning objectives for Object-Oriented Design and Programming I are listed below. Please note that each learner’s progress will be assessed at regular intervals using a variety of different assessment tools and techniques; in addition, learners will have the opportunity to self-assess their progress using Java and D2L online activities.

Table 1. CS 221 Learning Outcomes

Course Outcome	Assessment Activity
1. Given a description of a problem, apply the problem-solving steps used in computer programming to create a solution design.	Project
2. Working from a solution design, implement a solution to a problem using the Java programming language.	Project
3. Use incremental development to construct a working Java program.	Projects, online discussion/chat

Course Outcome	Assessment Activity
4. Identify and apply appropriate data types within a Java solution.	Quiz
5. Describe and identify key object-oriented programming concepts.	Quiz
6. Differentiate between the memory allocation approach for primitive and reference data types in Java.	Quiz
7. Examine the code available in the Java standard class libraries, and incorporate relevant Java standard classes into object-oriented design and program construction.	Project
8. Create and document program design solutions for simple Java programs.	Project, quiz
9. Given a solution design, create programmer-defined classes and incorporate these classes into Java program solutions.	Project, online discussion/chat
10. Distinguish among the options for input and output using Java, and select appropriate approaches for a given Java solution.	Project, quiz
11. Describe scope and persistence of objects and variables in object-oriented programming.	Quiz
12. Identify and correctly apply sequence, selection, and iteration/repetition patterns in object-oriented Java solutions and program designs.	Project, online discussion/data, quiz
13. Identify and apply advanced class and object features, including: overloading methods and constructors, argument passing, object return from methods, and organizing classes into packages.	Project, quiz
14. Manipulate collections of data using arrays and objects to solve a given problem using Java.	Project, quiz, online discussion/quiz
15. Describe the different sorting options available and select the best basic sort for use in a Java solution.	Project, quiz
16. Read and interpret UML 2.0 diagrams that document a problem, and implement the proposed solution using Java.	Project, quiz

Course Outcome	Assessment Activity
17. Implement professional standards and guidelines for designing and coding Java computer programs.	Project
18. Plan for and schedule adequate time to complete labs and projects no later than the required due date.	Observation by instructor, instructor review with student
19. Consult various online and independent resources to independently attempt to resolve problems BEFORE requesting assistance from co-workers/co-learners or supervisor/instructor.	Observation by instructor and tutor, instructor review with student
20. Determine when it is appropriate to seek assistance, from co-workers/co-learners or supervisor/instructor to resolve problems that could not be resolved independently.	Observation by instructor and tutor, instructor review with student

The learning objectives described in Table 1 relate directly to the Computer Science Program Objectives found at: http://www.uwosh.edu/departments/computer_science/cs_major/program_objectives.php

If you have a major or minor other than Computer Science, please consult the learning objectives for your specific program to determine the contribution of CS 221 learning objectives to your degree program.

DETERMINATION OF GRADES

The previous section on learning outcomes and assessment of student learning described what you can expect to learn in the course and how it will be assessed. The items that will contribute to your final grade and the percent of contribution are listed below:

	<u>Points Possible</u>
Projects – 5 @ 95 points	475
Online Quizzes – 4 @ 75 points	300
Online Chat Attendance/Participation 14 @ 7.5 pts	105
On-time Due Dates/Deadlines – 8 @ 15pts*	120
Total Possible Points:	1000

*All projects and online quizzes must be completed by the due day/time to receive an additional 15 points. Note that you may miss ONE due day/time without penalty – this should be used only under dire circumstances that prevent you from keeping up with your work in the class and from completing the assigned project or quiz.

Your letter grade for the course will be determined based on total points earned in the course as follows:

Point Range	Grade	Point Range	Grade
920 to 1000	A	720 to 789	C
890 to 919	AB	690 to 719	CD
820 to 889	B	600 to 689	D
790 to 819	BC	0 to 599	F

ACADEMIC HONESTY

“Integrity is doing the right thing, even if nobody is watching”

- Unknown

As an adult, you have the choice to complete the course requirements to the best of your abilities, or sacrifice your integrity and reputation for what you may perceive as a necessity at the time. If, at any point during the semester, you feel pressured to commit an act of academic misconduct in order to successfully fulfill course requirements, please contact me immediately. I’m sure we can come up with a plan of action that will help you succeed in the course and maintain your integrity. Please note that the following actions constitute academic misconduct and are subject to disciplinary action under the [UW Oshkosh Student Discipline Code](#) (2007): claiming the work of others as one’s own – whether the work is given willingly or unwillingly/unknowingly by another student, copied from an internet site of any kind contrary to course requirements, or originating anywhere other than as your own work product; cheating on an examination or gaining unauthorized access to examination materials; collaborating on work with others, contrary to the stated requirements of the course; failing to appropriately identify and cite the sources or ideas presented in a work product for the course; submitting work previously presented in another course; tampering with or destroying work completed by other students; or assisting other students in any of these acts. Students who feel compelled to engage in academic misconduct will be subject to the penalties defined in UWS Chapter 14 of the UWO Student Discipline Code.

COURSE CALENDAR

A complete course calendar and detailed activities for each week are posted in the D2L course site, which can be accessed at: <https://uwosh.courses.wisconsin.edu>.