

**University of Wisconsin Oshkosh
Computer Science Department**

**CS 221: Object-Oriented Design and Programming I
Fall 2008**

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

COURSE SCHEDULE

| | | Days | Time | Location |
|--------------------|---------|-------------|--------------|-----------------|
| Section 001 | Lecture | MW | 8:00-9:00am | HS 202 |
| | Lab | F | 8:00-9:00am | HS 101C |
| Section 002 | Lecture | MW | 9:10-10:10am | HS 208 |
| | Lab | F | 9:10-10:10am | HS 101C |

INSTRUCTOR

Dr. Robert (Bob) Ball

Computer Science Department

EMAIL ballr@uwosh.edu PHONE (920) 424-7080

IN-PERSON OFFICE HOURS*(HS 218): MTWRF 1:20pm-2:40pm

*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 course is a combined lecture/laboratory experience that maximizes learner involvement. The course will meet on MW in Halsey Science (HS) 208 classroom for demonstrations, team problem-solving, lecture, discussion, and related learning experiences; it will meet on F in the HS 101C teaching lab for hands-on design and programming experience. Occasionally we may spend more than one day per week in the teaching lab, or more than one day per week in the classroom setting working on problem-solving activities. Any changes to meeting times/locations will be posted as a news item on the course home page in D2L.

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.

If you feel intimidated by this list of things you are expected to know, you may wish to delay taking this course until next semester. Do consult with your academic advisor to determine how CS 115 and/or CS 142 would fit into your schedule instead; both of these classes will help better prepare you to succeed in CS 221. If you are unfamiliar with any of these items, please spend time reviewing how to accomplish these tasks. It is very important that you be familiar with them by the end of the first week of classes. In addition, some great sources of information are the Help feature of WindowsXP or MS Word. Information about using UWO email is available at:

http://www.acs.uwosh.edu/mail/titanmail/titanmail_faq.php and information about D2L is available at: <https://uwosh.courses.wisconsin.edu/>

MATERIALS AND RESOURCES

REQUIRED TEXTBOOK Wu, C. Thomas. (2006). An Introduction to Object-Oriented Programming with Java. McGraw-Hill Higher Education. ISBN: 0-07-294652-0.

SOFTWARE

NetBeans IDE 6.1 available in Campus Labs. Download free for your own computer from: <http://java.sun.com/javase/downloads/netbeans.html>. Select the **Java SE Development Kit 6u7 and NetBeans IDE 6.1 Bundle** to download both the NetBeans development environment and JDK6.

OTHER

USB Flash Drive with a minimum 512M capacity; 1G recommended. You will also have access to TitanFiles web-based storage.

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. | Classroom activity, lab observation, team presentation |
| 2. Working from a solution design, implement a solution to a problem using the Java programming language. | Classroom activity, Programming projects |
| 3. Use incremental development to construct a working Java program. | Lab observation, exam |
| 4. Identify and apply appropriate data types within a Java solution. | Programming projects, exam |
| 5. Describe and identify key object-oriented programming concepts. | Exam |
| 6. Differentiate between the memory allocation approach for primitive and reference data types in Java. | Exam |
| 7. Examine the code available in the Java standard class libraries, and incorporate relevant Java standard classes into object-oriented design and program construction. | Programming projects |
| 8. Create and document program design solutions for simple Java programs. | Programming projects, exam |
| 9. Given a solution design, create programmer-defined classes and incorporate these classes into Java program solutions. | Classroom activity, Programming projects |

| Course Outcome | Assessment Activity |
|---|---|
| 10. Distinguish among the options for input and output using Java, and select appropriate approaches for a given Java solution. | Exam |
| 11. Describe scope and persistence of objects and variables in object-oriented programming. | Exam |
| 12. Identify and correctly apply sequence, selection, and iteration/repetition patterns in object-oriented Java solutions and program designs. | Classroom activity, programming projects, exam |
| 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. | Programming projects, exam |
| 14. Manipulate collections of data using arrays and objects to solve a given problem using Java. | Classroom activity, programming projects, exam |
| 15. Describe the different sorting options available and select the best basic sort for use in a Java solution. | Classroom activity, programming projects, exam |
| 16. Apply test-first development to the construction of an object-oriented computer program. | Classroom activity, team presentation |
| 17. Read and interpret UML 2.0 diagrams that document a problem, and implement the proposed solution using Java. | Programming projects, exam |
| 18. Implement professional standards and guidelines for designing and coding Java computer programs. | Programming projects |
| 19. Present and justify, to a group of peers, the design and implementation of a problem solution. | Team presentation |
| 20. 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 |
| 21. 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 |
| 22. Determine when it is appropriate to seek assistance, from co-workers/co-learners or | Observation by instructor and tutor, instructor review with student |

| Course Outcome | Assessment Activity |
|---|---------------------|
| supervisor/instructor to resolve problems that could not be resolved independently. | |

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 Course Policies section described the importance of meeting due dates, and the different elements that will contribute to your overall grade. The items that will contribute to your final grade and the percent of contribution are listed below:

| | <u>Points Possible</u> |
|---|------------------------|
| Projects – 6 @ 50 points | 300 |
| Lab/Lab Quizzes - 10 @10 points | 100 |
| Classroom Activities – Variable based on activity | 100 |
| Exams (2 @ 150 points; 1 @ 200 points) | <u>500</u> |
| Total Possible Points: | 1000 |

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 |

COURSE POLICIES

You will be responsible for conforming to all course policies, related rewards, and consequences as they are documented here, whether or not they are explicitly discussed in class.

CLASSROOM ACTIVITIES

*“Eighty percent of success is showing up.”
-Woody Allen*

It is my goal to make the scheduled meetings so interesting that you don’t want to miss one! But please keep in mind that object-oriented design and programming require the development of analytical and technical knowledge and skills. The skills and knowledge taught in this course cannot be acquired from reading a textbook and taking an exam. You must be actively involved and engaged in the learning process to achieve the desired learning outcomes and be prepared for your future coursework and career.

Many in-class and in-lab learning experiences that actively involve learners will be included in the scheduled course meetings. Attendance in all scheduled meetings, just like attendance in future professional meetings, is required. It is my job to make the classes and labs interesting enough to be worth your attendance! Also note that I based the course loosely on the textbook, and will be supplementing that material with other content that will be provided in class sessions.

When the classroom activity involves participation that produces a work product – a design diagram, program code, a mini-presentation, or other form of participation – you will be asked to submit the work product for evaluation. Material submitted will be assigned a point grade based on: quality of work produced, demonstrated understanding of the activity and related subject matter, and other factors relevant to the activity. Note that you must be present to earn classroom activity points – if you choose to miss a class session that includes an activity you may not make up the points, unless your absence is approved in advance by the instructor or you provide an excuse from a professional health care provider. Your classroom activity points constitute 10% of your overall course grade.

The bottom line is that you are expected to be in scheduled sessions because active involvement in the learning process is key to your successful completion of the course. You are an adult and free to make choices about your attendance within and aside from the parameters and consequences described above.

PREPARATION

“To be prepared is half the victory.”
- Miguel de Cervantes Saavedra

In addition to attending scheduled sessions, the successful learner must be prepared. The D2L course site lists reading projects, required and optional learning activities, lab instructions, due dates, and other learning tools. It is the responsibility of the learner to complete scheduled items prior to the class period and be prepared to engage in active learning exercises.

As stated in my “Letter to the Student”, you can reasonably expect to spend three hours outside of class for each 1 credit of class – meaning 9 hours per week *in addition to* class/lab attendance. This should allow you to be both prepared for class sessions, and complete all projects – though programming is a difficult activity for which to estimate time and on occasion additional time may be required. You should plan accordingly, and schedule your time wisely.

DUE DATES *“A perfect method for adding drama to life is to wait until the deadline looms large.”*
-Unknown

Object-oriented design and programming is a cumulative topic – meaning you can’t skip a part or set of concepts and go on to the next without doing serious damage to the learning process. Consequently, once you get behind in the class it is difficult to get caught up! To help ensure your success, you must complete all projects and programs by the stated due dates. Again, because the knowledge and skills you’ll be learning will require you to actively engage with the material, you will need to plan sufficient time to complete projects and programs. And as previously mentioned, estimated time to complete a program is difficult. One program may go quickly, the next one may require far more time so do plan ahead and don’t wait until the last possible moment to begin your work. Also note that you will need to

commit to spending more time on labs and projects than just the weekly scheduled class time – plan time each week to spend on your programming labs/projects.

Please note that NO LATE LABS, PROJECTS, PROGRAMS, or OTHER ASSIGNMENTS will be accepted and those that are submitted late will receive NO points ; however, you may hand in incomplete/partially complete items for partial credit by the designated due date/time. Do not expect to do well in the course if you do not stay caught up or if you hand in only partially completed programs. Due dates will be summarized as a news item on the course home page in D2L, provided with all instructions, and posted on the corresponding D2L dropbox. You will have sufficient time to plan your schedule. If you have special needs or encounter circumstances that will prevent you from meeting a due date, please contact me BEFORE the scheduled due date so we can establish a plan to meet your needs and avoid your potential loss of points.

Also note that I have numerous office hours during the week, including virtual office hours, and I can be reached by email. I am happy to help you succeed in the course so don't be afraid to ask for assistance when needed. In addition, a tutor will be available to assist you during regularly scheduled hours (hours will be posted on D2L when scheduled). Don't let being frustrated or getting "stuck" keep you from meeting a due date or deadline.

PROJECTS *"If your project doesn't work, look for the part that you didn't think was important"*
-Arthur Bloch

Object-oriented design and programming labs and projects are the primary learning activity in which you will engage in the course. Each project will require BOTH the design of a program, which will be documented in diagrammatic form, and a working Java program. Projects will have multiple components – some of which will be completed during scheduled lab times and other that will be completed outside of scheduled class times. Criteria for assessing projects will be provided and will provide a roadmap for required activities.

Many of the projects will be team projects – you will choose one other person in the class as a partner. Note that it is important that you complete the project working alone as a team – you may not consult with other students/teams in the course. You must make every effort to solve your problems and work through the design and programming learning process with the material provided to you throughout the class. When you need assistance, rather than seek out someone who is unfamiliar with the instructional techniques used in this specific course, you should see the designated tutor or the instructor. Any submitted work that too closely represents the work of another team will be closely scrutinized and may impact your project grade. Please note that your project scores are worth 30% of your overall grade in the course. You will work on six different projects at 50 points per project.

EXAMS *"Examinations are formidable even to the best prepared, for the greatest fool may ask more than the wisest [person] can answer."*
-Charles Caleb Colton

In addition to demonstrating skills/abilities and professional attitudes/behavior in the course, each student must demonstrate an understanding of key concepts and elements in object-oriented design and programming. This understanding extends to use of key terms and design notations, and the ability to

read and interpret the designs and code of others. These aspects of the learning process will be assessed in three exams, the first two worth 150 points each and the third exam worth 200 points. All three exams will combine to contribute to 50% of your overall grade for the course. Exams will be conducted during the regularly scheduled lab session in the HS101C lab on the following dates:

Exam I: Friday, Oct 3
Exam 2: Friday, Nov 7
Exam 3: Friday, Dec 12

If you have any time conflicts with these dates, please let the instructor know as soon as possible. All exams will use the D2L Quiz platform. A combination of different types of problems will be included. A sample exam will be available for students to practice with prior to the first exam. This will allow students to gain familiarity with the testing platform and know what to expect during the exam.

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 see 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.