1 Prerequisites and description

This course is 4 credits. Its prerequisites are: a C or better in CMSC 131 (or a score of 5 on the A Java AP exam, or passing the CMSC department’s placement exam), and a C or better in MATH 140.

Object–Oriented Programming II is a second programming course for computer science and computer engineering majors, which is the second part of a two–semester introduction to object–oriented programming. The course provides an introduction to the use of computers to solve problems using software engineering principles. In it students design, build, test, and debug medium–sized software systems, and learn to use relevant tools. The course involves use of object–oriented methods to create effective and efficient problem solutions. Students will use and implement application programming interfaces (APIs). Programming is done in Java.

2 Contact information

2.1 Email contact

We’re not able to explain most course material via email. It is more appropriate for class discussion or personal communication, and this is also due to the size of the course. Also please discuss regular course business with us, including administrative issues, in person when possible (before and after class are good times), and use email in case of urgent or emergency matters.

Due to time constraints and other factors it is not practical to provide detailed information or assistance regarding programming projects via email, and attempting to do so often results in students receiving incomplete or inadequate information. Therefore please ask questions about projects in person, either during office hours, or before or after class.

2.2 Instructors

<table>
<thead>
<tr>
<th>Sections 010X, 020X, and 030X</th>
<th>Section 0101H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor: Larry Herman</td>
<td>Instructor: Dr. Jim Purtilo</td>
</tr>
<tr>
<td>Office: 1111 A. V. Williams</td>
<td>Office: 4437 A. V. Williams</td>
</tr>
<tr>
<td>Phone: (301) 405–2762</td>
<td>Phone: (301) 405–2706</td>
</tr>
<tr>
<td>Email: <a href="mailto:larry@cs.umd.edu">larry@cs.umd.edu</a>(*)</td>
<td>Email: <a href="mailto:purtilo@cs.umd.edu">purtilo@cs.umd.edu</a>(*)</td>
</tr>
</tbody>
</table>

(*) See Section 2.1 above regarding email.

Office hours will be provided in a separate handout shortly.

2.3 Teaching assistants

Information about the TAs, and their office hours, will be provided in a separate handout shortly.

While the TAs will provide assistance with assignments and projects during office hours, you are ultimately responsible for developing and debugging your own homework assignments and projects, which are your coursework and you’re receiving a grade for. You should not rely on the instructional staff to get your projects to work. If you come to office hours for help with project or homework debugging you may be guided in the right direction, after which you would be expected to resume attempting to solve the problem on your own.

3 Class webpage

Many course materials will be made available via the class webpage at [www.cs.umd.edu/class/fall2010/cmsc132](http://www.cs.umd.edu/class/fall2010/cmsc132). Accessing the webpage will require an ID and password to be provided in class.

4 Textbooks


The second edition of this text is also acceptable (2010, ISBN 978–0–470–12870–1).
Both texts are recommended rather than required.

5 Course evaluations

Course evaluations are important, and the department and faculty and instructors take student feedback seriously. The system may be accessed near the end of the semester at www.courseevalum.umd.edu. However, and more importantly, rather than waiting until the end of the semester to give feedback, please bring any questions or concerns to our attention during the course; preferably in person if at all possible. An instructor cannot guarantee to be able to solve any problem or change anything that is discussed, but can’t do anything if any issues that arise are not brought to their attention, so we welcome hearing any comments, questions, suggestions, or concerns that you may have.

6 Course topics (SUBJECT TO CHANGE)

The following list of topics may vary according to the pace of lecture, so their order and duration are approximate.

Week #1: Introduction to object–oriented programming (OOP), Java support for OOP, Java language features
Week #2: Abstract classes, Java support for OOP, program testing, program correctness
Week #3: Inner classes, event–driven programming (GUI introduction)
Week #4: Algorithmic complexity, Java collections, linear data structures
Week #5: Linear data structures, con’t.
Week #6: Generic programming, sets, maps, hashing
Week #7: Recursion, trees, binary search trees, heaps, priority queues
Week #8: Trees, con’t.
Week #9: Java I/O, network basics, software development
Week #10: Software development, software lifecycle, software process models, problem specification and design, object–oriented design
Week #11: Unified Modeling Language (UML), threads and synchronization
Week #12: Graphs, graph traversals, shortest paths
Week #13: Sorting, algorithm strategies
Week #14: Design patterns
Week #15: Effective Java

7 Computing resources

All assignments can be done on the machines of your choice. The Office of Information Technology has several computing labs on campus, for those who need to use them; see www.oit.umd.edu/wheretogo. You are welcome to do your work on a home computer if you have one. There should not be any machine–specific dependencies in your code. However, if we are not able to run your program because there is a difference between your computer environment and ours, you are responsible to work with us to get your program to work in our environment.

8 Attendance and grades

Grades will be recorded on the CMSC lower–level course grade server at https://grades.cs.umd.edu and may be checked for correctness there.

Students are responsible for all academic and administrative material discussed in lecture and discussion section, regardless of whether they were in class to hear the information or not. It’s understood that students may occasionally miss class for various reasons, but email and office hours are not intended as a replacement for class attendance. A student who was not class for any reason will need to find out what was missed and to get notes from a classmate who was present.

Coursework will count toward the final grade according to the following percentages:

- **Midterm:** one midterm expected 20%
- **Final:** will be comprehensive 25%
- **Programming projects:** five expected coding assignments 35%
- **Quizzes:** in discussion section 10% (equally weighted)
- **Homeworks** short programming exercises 10% (equally weighted)
The project submission and grading policies will be provided when the first project is assigned. All projects will be graded out of 100 points, but depending upon their relative difficulty, which can't be predicted in advance, they won't be weighted equally. Their weights will be given at the end of the semester.

Any request for reconsideration of the grading on any coursework must be submitted within one week of when it is returned. Exam regrading requests must be made in writing. Any coursework submitted for reconsideration may be regraded in its entirety.

Final course grades will be curved as necessary, based on each student’s total numeric score for all coursework at the end of the semester. However, since the grade distribution won’t be known until all coursework is graded, whether there is a curve, or what the letter grade ranges might be, can’t be predicted in advance.

9 Quiz, exam, and final dates

Quizzes will be given in discussion section and will cover discussion and lecture material. They will be announced in the prior discussion section or lecture.

The midterm exam will be held during lecture. The midterm date will be confirmed later, and may vary depending on lecture progress and other factors.

The final exam date and time will be rescheduled only for students having another final at exactly the same time, or for students with more than three final exams scheduled on the same day. (The only courses that students who are enrolled in CMSC 132 should be able to take that have finals at the same time as its final are BMGT 221 and ENES 221.) If either of these situations applies to you, you must inform your instructor at least two weeks in advance of the final exam time for any allowances to be made. Also please inform your instructor immediately if you have a conflict with the scheduled midterm date, or any other important date as the semester progresses.

<table>
<thead>
<tr>
<th>Midterm</th>
<th>Final exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wednesday, October 20</td>
<td>Tuesday, December 14, 4:00 p.m.–6:00 p.m.</td>
</tr>
</tbody>
</table>

10 Absences and accommodations

Besides the policies in this syllabus, various University policies may apply to students during the semester. Policies that may be relevant appear in the Undergraduate Catalog, at www.umd.edu/catalog.

If you experience difficulty during the semester keeping up with the academic demands of your courses, you may consider contacting the Learning Assistance Service in 2201 Shoemaker Building at (301) 314–7693. Their educational counselors can help with time management issues, reading, note-taking, and exam preparation skills.

10.1 Excused absences

Missing an in–class assessment (a quiz or an exam) for reasons outside of your control (such as illness, religious observance, participation in required university activities, or family or personal emergency such as a close relative’s funeral or serious accident) will be considered to be an excused absence. Students requesting an excused absence must furnish documentary support of the cause of the absence. For an absence due to medical reasons documentation would be from a health care professional who treated you. Excused absences will not be given unless documentation is provided.

In cases of illness the documentation must show that you were treated by a health professional, and that in their judgment you were incapacitated and therefore unable to attend, for an absence to be considered excused. The documentation must include the phone number of the health professional, and indicate the exact dates or times of incapacitation, which must include the date of the missed assessment.

It is the University’s policy to provide accommodations for students with religious observances conflicting with in–class assessments, but it is the student’s responsibility to inform the instructor in advance of intended religious observances that will (or may) conflict.

An excused absence for an exam will be handled either by averaging the student’s scores for the other exams (possibly a weighted average), or by giving a makeup exam. In the case of an excused absence for a quiz no makeup will be given; the score will be computed as the average of the student’s scores for the other quizzes at the end of the semester.

A student who might miss an in–class assessment for any reason other than circumstances outside of their control must contact the instructor in advance (or as soon as possible) to discuss the reason. According to the University policy, an instructor is not obliged to make allowances other than for reasons such as those discussed above.

The policies for excused absences above do not apply to project assignments. In cases of extremely serious, lengthy, documented illness or other protracted, severe emergency situations, the instructor may consider extensions on project assignments, depending upon the circumstances.
10.2 Students with disabilities

Students with disabilities who have been certified by Disability Support Services as needing any type of special accommodations should see the instructor as soon as possible, during the schedule adjustment period.

All arrangements for exam accommodations as a result of disability must be made and arranged with the instructor at least three business days prior to the exam date, or accommodations cannot be made.

11 Academic integrity statement

The Campus Senate has adopted a policy asking students to include the following statement on each examination or assignment in every course: “I pledge on my honor that I have not given or received any unauthorized assistance on this examination (or assignment).” Consequently, you will be requested to include this pledge on each exam and project.

Please carefully read the Office of Information Technology’s policy regarding acceptable use of computer accounts and resources at www.nethics.umd.edu/aup.

Programming projects are to be written individually, so cooperation or use of unauthorized materials on projects is a violation of the University’s Code of Academic Integrity. Any evidence of this, or of use of unauthorized materials or cooperation on exams or quizzes, or other possible violations of the Honor Code, will be submitted to the Student Honor Council, which could result in an XF for the course, suspension, or expulsion.

- In learning the course concepts students are welcome to study together or to receive help from anyone else. Students may discuss with others the project requirements, the C language, what was discussed in lecture and discussion section, and general debugging or syntax errors.

- When it comes to actually designing, coding, or debugging a project assignment, other than help from the instructional staff a project must solely and entirely be a student’s own work. Working with another student or individual, or using anyone else’s work in any way except as noted below, is a violation of the code of academic integrity and will be reported to the Honor Council. Questions such as “How did you implement this part of the project?” or “Please look at my code and help me find my stupid error!” exemplify disallowed cooperation.

Violations of the Code of Academic Integrity may include, but are not limited to:

1. Failing to do any of the work on a project by yourself, other than assistance from the instructional staff.
2. Using any ideas or any part of another person’s project, or copying anyone else’s work in any way.
3. Giving any parts or ideas from your project, including test data, to another student.
4. Allowing any other students access to your program on any computer system.
5. Transferring any part of a project to or from another student or individual by any means.

When designing or writing projects students are free to use information and code provided by the instructional staff, only if the source is cited in a comment in the relevant section of the program, and the substantial part of a student’s program remains their own individual work.

If you have any question about a particular situation or source then consult with the instructor in advance. Should you have difficulty with a programming assignment please see the teaching assistants in office hours – do not solicit help from anyone else in violation of these rules.

IT IS THE RESPONSIBILITY, UNDER THE HONOR POLICY, OF ANYONE WHO SUSPECTS AN INCIDENT OF ACADEMIC DISHONESTY HAS OCCURRED TO REPORT IT TO THE INSTRUCTOR, OR DIRECTLY TO THE HONOR COUNCIL.

Students are welcome and encouraged to compare or discuss with others their implementations of programming projects after they are graded, provided that none of the students in question still have to successfully submit that project assignment, and only if it has been announced that that project will not be extended upon in a later project.

12 Right to change information

Although every effort has been made to be complete and accurate, situations that arise during the semester could require the adjustment of any material given here. Consequently, given due notice to students, the instructor reserves the right to change any information or policies in this syllabus or in other course materials.

13 Copyright

All course materials are copyright Larry Herman and Jim Purtilo (and other CMSC faculty and instructors) © 2010. All rights reserved. Students are permitted to use course materials for their own personal use only. Course materials may not be distributed publicly or provided to others (excepting other students in the course), in any way or format.