EAS 240: Introduction to Programming for Engineers  
Syllabus – Spring 2018

Basic Information:

EAS 240: Introduction to Programming, 3 credits  
**Lecture Schedule:** TR, 12:30 PM – 1:50 PM  
**Lecture Location:** Knox 109  
**Prerequisite:** MTH 141; Approved and Intended Engineering Majors Only  
**Piazza/UBlearns:** The syllabus, lectures, course announcements, programming exercises and solutions, projects, practice exams, and Q&A will be accessible through Piazza/UBlearns

Instructor:

**Dr. Pedram Johari**  
Email: [pedramjo@buffalo.edu](mailto:pedramjo@buffalo.edu) (*Please put “EAS 240” in the subject of your emails*)  
Office: 230C Davis Hall  
Office Hour: Tuesday 3 PM – 5 PM

Teaching Assistants:

**Britton Medley**  
Email: bjmedley@buffalo.edu  
Office Hour: Tuesdays & Thursdays 5:00PM – 6:00PM in 204 Davis Hall or by appointment.  

**Mikhail Shalaev**  
Email: mshalaev@buffalo.edu  
Office Hour: Thursdays 3:00PM – 5:00PM in 117 Davis Hall or by appointment.  

**Xianglong Miao**  
Email: xmiao2@buffalo.edu  
Office Hour: Fridays 11:00AM – 1:00PM in 230V Davis Hall or by appointment.  

**Nikhilesh Sharma**  
Email: nsharma9@buffalo.edu  
Office Hour: Mondays and Wednesday 11:00AM – 1:00PM in 204 Davis Hall or by appointment.

Student Assistants:

**Maxwell McManus**  
Email: memcmanu@buffalo.edu

**Keith Sante**  
Email: kdsante@buffalo.edu

**Yizheng Li**  
Email: yizhengl@buffalo.edu

**Hussam**  
Email: habdella@buffalo.edu
EAS 240: Introduction to Programming for Engineers

Textbooks:


The required textbook is available in the campus book store.

Lectures:

Lectures are from 12:30 PM – 1:50 PM on Tuesday and Thursday in Knox 109. It is strongly recommended that you attend all lectures. It is expected that you will check your email and Piazza/UBlearns daily for course information.

The course does not have a recitation.

Catalog Description:

An introductory programming course for students in engineering and the sciences. The course covers the basics of procedural and object oriented programming; and in particular basic programming concepts such as statements, conditions, functions, pointers, I/O, objects, data structures, and algorithms. The course will include exercises focusing on applications for solving scientific and engineering problems.

Catalog Requirements:

During the course, it is expected that students will complete the following work. The quantity and timing of material is subject to change based on course progress.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Material</th>
<th>Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 programming exercises</td>
<td>4 exercises on procedural programming; 4 on object-oriented programming.</td>
<td>Weeks 2,3,4,5; Weeks 9,10,11,12</td>
</tr>
<tr>
<td>Individual project</td>
<td>Procedural programming</td>
<td>Week 6</td>
</tr>
<tr>
<td>Group project</td>
<td>Object-oriented programming</td>
<td>Week 13</td>
</tr>
<tr>
<td>Midterm exam</td>
<td>Procedural programming</td>
<td>Week 7</td>
</tr>
<tr>
<td>Final exam</td>
<td>All material</td>
<td>Week 15</td>
</tr>
</tbody>
</table>
Course Learning Outcomes and Assessment Tools:

Upon successful completion of the course, students should be able to:

<table>
<thead>
<tr>
<th>Course Learning Outcomes</th>
<th>Assessment Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Understand the fundamental constructs used in procedural programming, including variables, data types, arrays, loops, conditionals, functions, and data input/output.</td>
<td>Programming exercises 1,2,3,4; Midterm exam; Final exam</td>
</tr>
<tr>
<td>2 Write, debug, and test a procedural program.</td>
<td>Programming exercises 1,2,3,4</td>
</tr>
<tr>
<td>3 Design and implement computer algorithms to solve engineering problems using procedural programming.</td>
<td>Programming exercises 3,4; Project 1</td>
</tr>
<tr>
<td>4 Understand the fundamental constructs used in object oriented programming, including objects, properties, methods, interfaces, and polymorphism.</td>
<td>Programming exercises 5,6,7,8; Final exam</td>
</tr>
<tr>
<td>5 Write, debug, and test an object oriented program.</td>
<td>Programming exercises 5,6,7,8</td>
</tr>
<tr>
<td>6 Design and implement computer algorithms to solve engineering problems using object oriented programming.</td>
<td>Programming exercises 7,8; Project 2</td>
</tr>
<tr>
<td>7 Read, interpret, and understand the operation of an algorithm written by others.</td>
<td>Midterm exam; Final exam</td>
</tr>
</tbody>
</table>

The expected student learning outcomes listed above contribute directly to the Electrical Engineering Department’s Program Educational Objectives, which can be found at the following link: http://engineering.buffalo.edu/electrical/academics/bachelors_ee.html

Course Grading:

Your grade will be determined as follows:

<table>
<thead>
<tr>
<th>Item</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming exercises</td>
<td>24%</td>
</tr>
<tr>
<td>Individual project</td>
<td>10%</td>
</tr>
<tr>
<td>Group project</td>
<td>10%</td>
</tr>
<tr>
<td>Midterm exam</td>
<td>20%</td>
</tr>
<tr>
<td>Final exam</td>
<td>30%</td>
</tr>
<tr>
<td>Professionalism</td>
<td>6%</td>
</tr>
</tbody>
</table>
Grades will be assigned as in the table below:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Score</th>
<th>Grade</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>[90-100]</td>
<td>C</td>
<td>[66-70]</td>
</tr>
<tr>
<td>A-</td>
<td>[86-90)</td>
<td>C-</td>
<td>[62-66)</td>
</tr>
<tr>
<td>B+</td>
<td>[82-86)</td>
<td>D+</td>
<td>[58-62)</td>
</tr>
<tr>
<td>B</td>
<td>[78-82)</td>
<td>D</td>
<td>[54-58)</td>
</tr>
<tr>
<td>B-</td>
<td>[74-78)</td>
<td>F</td>
<td>[0-54)</td>
</tr>
<tr>
<td>C+</td>
<td>[70-74)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I reserve the right to lower the cutoffs (i.e., give higher grades than indicated) so that the overall class average will not be lower than a B-. However, I will not raise the cutoffs.

**End-of-semester grade:** Your end-of-semester grade is the grade that you earned based on your performance throughout the entire course. Your final grade cannot be changed unless the posted grades on UBlearns do not match the actual grades that you received (i.e., the wrong score is listed for an exam or other assignment). It is against university policy for me to provide individual students opportunities to improve their grades at the end of the course that I do not provide to every student. Therefore, if you are worried about your performance/grade, please meet with me as soon as possible and, importantly, before you take the final exam.

**Incompletes (I/IU):** A grade of incomplete (“I”) indicates that additional work is required to fulfill the requirements of a given course. Students may only be given an “I” grade if they have a passing average in coursework that has been completed and have well-defined parameters to complete the course requirements that could result in a grade better than the default grade. An “I” grade may not be assigned to a student who did not attend the course.

The instructor will follow the incomplete policy as it is described on the following website: [http://undergrad-catalog.buffalo.edu/policies/grading/explanation.shtml#incomplete](http://undergrad-catalog.buffalo.edu/policies/grading/explanation.shtml#incomplete)

**Course Tools:**

**Linux Mint:** We will use a Linux environment to code, compile, and execute your programs. Lecture 2 will provide a tutorial introduction to the Linux Terminal and describe several ways you can access Linux. For Windows users, we have prepared a virtual machine that you can download at UB Box and can run using VMware Workstation Player. Both can be accessed for free at the following links:

- Linux Mint VM: [https://buffalo.box.com/s/7j842donkbsdw44ad672s4o1sm8wv4k9](https://buffalo.box.com/s/7j842donkbsdw44ad672s4o1sm8wv4k9)

**Autolab grading system:** We will use an automatic grading system called Autolab, which was developed at Carnegie Mellon University (CMU). The Autolab installation that we will access is deployed and maintained by the Department of Computer Science and Engineering (CSE) at UB. This system will give you immediate feedback on your homework grade and will allow you to repeatedly submit your solutions until the deadline. It also provides a real-time scoreboard showing
how well you have done on an exercise compared to every other student in the class (this scoreboard uses anonymous usernames for privacy). Autolab can be accessed at the following link:

https://autograder.cse.buffalo.edu/

You will receive an email from the UB autograder system inviting you to log in and change your password. If the password reset option has timed-out, just type in your UB email address and click “Forgot your password?” to gain access to your account.

**Piazza:** We will use Piazza (along with UB/learns) to post course content (syllabus, lectures, programming exercises, projects, and practice exams), to post course announcements, and for Q&A. The system is designed to get you help fast and efficiently from the instructor, TAs, and classmates. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. This will allow the course to be run more efficiently: the instructor and TAs will not have to answer the same question from multiple students; your questions may have already been answered on Piazza; or one of your peers may be able to answer your question before the instructor or TAs.

If you send the instructor or TAs an email that would be best addressed publicly on Piazza, we will ask you to post it on Piazza. If you have to communicate something privately to the instructor or TAs, you may do this on Piazza or by email. If you have any problems or feedback for the developers, email team@piazza.com. Our class page can be accessed at the following link:

https://piazza.com/buffalo/spring2018/eas240/home

We ask that you please do not post your complete solutions to programming exercises or projects on Piazza. This is not an effective way to help your peers and it is against our academic honesty policy.

**Programming Exercises:**

**Assignments:** Programming exercises will be assigned during the weeks outlined in the Catalog Requirements section (above). Homework will be submitted electronically and must be submitted before the posted time on the submission system. Late homework will not be accepted by the electronic submission system and will not be accepted by the instructor or the TAs. Due to this strict policy, your two lowest programming exercise grades will be dropped. Homework solutions will be posted after the deadline.

**Acceptable forms of cooperative learning:** You are encouraged to join with other students in discussing the course, including homework. The most fruitful (and the only acceptable) forms of collaboration are to:

- Try to solve the problems on your own and, if your results differ, focus on understanding the reasons for any differences between your answer and someone else’s approach, rather than just copying the answer that someone else got.
- Discuss with your classmates possible approaches to solving the problems, and then have each one fill in the details and write her/his solution independently.

**When you write up the work that you hand in, you should do so on your own. It is unacceptable to copy a solution that someone else has written.**
EAS 240: Introduction to Programming for Engineers

Programming Projects:

There will be two larger scale programming projects given in the course. The first one you will do individually and the second one you will do in a group, where each person will have a defined part to work on. It is through larger scale projects like this that you will learn the most about programming and have the most fun.

Lecture Format:

Lecture will be primarily taught via computer projector. You can always come to class with a printed copy of the lecture slides and a notebook (or additional paper) for taking notes, because I sometimes cover more than is in the notes. I expect you to read the lecture slides before class and the relevant sections from the textbook so that you can get the most out of lecture. It should be noted that attending class will help you learn the material better and in less time than if you were to attempt to learn it entirely on your own.

Exams:

There will be one midterm exam and one final exam (cumulative). All exams will be taken in class and will be closed book and closed notes. However, you are allowed to bring one double-sided 8½” x 11” “cheat sheet” for each exam. Makeup exams will be administered only under extenuating circumstances with official documents, provided that the instructor is notified in advance or there is an immediate medical or family emergency.

Professionalism:

Professionalism is how engineers practice and conduct themselves in their work and as individuals in society. In order to improve the working and learning environment within SEAS, and to best equip our students for employment after graduation, faculty are encouraged to include “Professionalism” as a graded component in their classes. Professionalism will be a small component of your grade (6%) and will be determined based on a few simple criteria listed below. It is recommended that you follow these guidelines in all classes and professional interactions.

- **Email:** Emails to faculty, staff, and TAs should be written formally with proper salutations and closings, full sentences, and correct spelling (no “txtspeak”).
  - Acceptable salutations: Hi/Dr./Prof./Professor X
  - Unacceptable salutations: Hey, Yo, Sup, etc.
  - Acceptable closings: Thank you, Thanks, Sincerely, Best, Best Regards, Cheers
- **Online conduct:** You must conduct yourself professionally on Piazza/UBlearns and in Autolab
  - Your usernames should not contain profanity, derogatory language, references to drugs, etc.
  - Do not use language that is – or could be construed to be – offensive toward others. Avoid derogatory and/or sarcastic comments directed at an individual or group of people.
EAS 240: Introduction to Programming for Engineers

- Do not type in ALL CAPS unless it is part of a code snippet where all caps is appropriate (e.g., for symbolic constants, NULL, EOF, FILE).
- Do not make fun of people’s questions or answers. All discussion should be constructive, well-meaning, and well-articulated (to the best of your ability).

- **Academic honesty:** If we discover that you have violated the academic honesty policy on an exercise, project or exam, you will not only get a 0 for that work, but also receive a 0 for your professionalism grade.

Your Professionalism grade will be out of 100 points. You will start with the full 100 points. 5 points will be deducted for each minor violation. Up to 100 points could be taken off for major violations in the online conduct rules. 100 points will be deducted for academic honesty violations.

**Office Hours:**

Please cooperate with us in respecting the office hours of the TAs and the professor. If the posted office hours are insufficient or inconvenient, then individual appointments can be made with either the instructor or TA. You may use Piazza at any time to discuss your questions, or contact the instructor/TAs/SA privately (through Piazza or email) to set up an appointment. We will respond as soon as possible, but you should be aware that we do not always check emails on the evenings and weekends, or while we are traveling. Please be on time to any appointments that you make.

**Student Feedback:**

Students are encouraged to use office hours, email (to the instructor or a TA/SA), or anonymous means to voice their concerns on teaching style and course material.

**Special Needs and Circumstances:**

Students with special needs or circumstances should feel free to meet with or contact the instructor for help as soon as possible.

If you have any disability which requires reasonable accommodations to enable you to participate in this course, please contact the Office of Accessibility Resources, 60 Capen Hall, 645-2608, and also the instructor of this course. The office will provide you with information and review appropriate arrangements for reasonable accommodations. [http://www.student-affairs.buffalo.edu/ods/](http://www.student-affairs.buffalo.edu/ods/)

**Academic Integrity:**

Academic honesty is essential to the intellectual life of the university. Academic honesty violations diminish the value of your degree and your peers’ degrees. Students who pass off the work of another person as their own are guilty of academic dishonesty. In addition to such acts of cheating or plagiarism, any unauthorized possession of examinations or tampering with grade records are acts of academic dishonesty specifically forbidden by the university’s rules. If we discover that you have violated the academic honesty policy on an exercise, project or exam, you will not only get a 0 for that work, but also receive a 0 for your professionalism grade.
EAS 240: Introduction to Programming for Engineers

For more information please consult:
http://undergrad-catalog.buffalo.edu/policies/course/integrity.shtml

Other Resources:
The course will be self-contained, however, additional resources are useful for reviewing material, getting a different perspective on a topic, or exposing you to additional problems. A few good resources, aside from the required books, are listed below:

MIT OpenCourseWare: 6.087 Practical Programming in C
- This course covers C programming using the textbook by Kernighan and Ritchie.
- My lectures are, in part, based on the lectures from this course.

C Programming tutorials at TutorialsPoint:

Resources related to Bjarne Stroustrup’s C++ textbook
- http://www.stroustrup.com/Programming/

Linux Terminal Command Reference and Tutorial from the Linux Mint community
- Tutorial: https://community.linuxmint.com/tutorial/view/100
- Command reference: https://community.linuxmint.com/tutorial/view/244
- We will teach you a small subset of commands, which should be sufficient for you to do basic file management tasks, navigate directories, and compile and execute C and C++ programs using the terminal.
- Most of these commands will also work in the Terminal application available on all Apple computers using Mac OS X.

Tentative Course Schedule:

Part 1: Procedural programming (programming will be done in C)
Week 4: File I/O. Debugging.
Week 6: Structures.
Week 7: Midterm.
Week 8: Spring break. No class.

Part 2: Object-oriented programming (programming will be done in C++)
Week 9: Introduction to C++. I/O.
Week 11: Interfaces and polymorphism.
Week 12: Data structures: vectors, stacks, and queues.
EAS 240: Introduction to Programming for Engineers

Week 13: Object libraries.
Week 14: GUIs and visualization (data representation using GNU Octave or MATLAB).
Week 15: Case study of a large program.
Week 16: Final exam.