Introductions

- **Instructor**
  - Prof. Dola Saha, PhD University of Colorado Boulder
  - [http://www.albany.edu/faculty/dsaha/](http://www.albany.edu/faculty/dsaha/)
  - dsaha@albany.edu

- **Students**
### Information

- **Course Website:**

- **Blackboard:**
  - [https://blackboard.albany.edu/](https://blackboard.albany.edu/)

<table>
<thead>
<tr>
<th>Course Website</th>
<th>Blackboard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture Slides</td>
<td>Homework Assignments</td>
</tr>
<tr>
<td>Class Calendar / Schedule</td>
<td>Homework Submission</td>
</tr>
<tr>
<td>Other Information</td>
<td>Homework Solutions</td>
</tr>
<tr>
<td></td>
<td>Announcements</td>
</tr>
<tr>
<td></td>
<td>Grades</td>
</tr>
</tbody>
</table>
Office Hours

- Where: BA 312

- When:
  - Tuesday 1:15-2:15PM
  - Thursday 2:00-3:00PM
  - By appointment.
Textbooks

Required:
- "C How to Program, 8th Edition", Paul Deitel and Harvey Deitel, Pearson

Reference:
- "The C Programming Language", Brian W. Kernighan and Dennis Ritchie, Pearson
- "Problem Solving and Program Design in C", Jeri R. Hanly and Elliot B. Koffman, Pearson
Assignments & Grading

- Assignments
  - No late assignments will be accepted.
  - All assignments are due by 11:59PM on the due date in Blackboard.
  - Re-grading requests will be considered up to 5 business days after posting the grades for the corresponding assignment.

- Grading
  - Attendance and class Participation - 5%
  - Class Assignments - 15%
  - Homework Assignments - 40%
  - Midterm 1 - 10%
  - Midterm 2 - 10%
  - Final Exam - 20%
Academic Integrity

- Undergraduate Academic Regulations
  - [http://www.albany.edu/undergraduate_bulletin/regulations.html](http://www.albany.edu/undergraduate_bulletin/regulations.html)

- Academic Dishonesty
  - Plagiarism, Cheating on examinations, unauthorized collaboration, etc.

- Practicing Academic Integrity
  - Citation

- Penalties for Violation
  - Warning, lowering grade, failing grade
In Class Decorum

- No use of phones
- No use of Computers / laptops
- Computers will be used only when directed in the class
- No crosstalk
- No Food/Drink
- Raise hand to ask questions
Why this course?
By end of the semester

- Demonstrate basic proficiency in the C programming language.
- Formulate algorithms to solve basic computational problems.
- Construct larger programs by identifying and solving sub-problems.
- Apply basic concepts of software engineering.
- Apply pointers, arrays, and structures correctly.
- Apply dynamic memory allocation correctly.
- Apply basic I/O operations to read and write data files.
- Understand the basic concepts of algorithmic complexity.
- Apply basic architectural concepts to program design.
# Course Calendar

<table>
<thead>
<tr>
<th>Date</th>
<th>Discussion Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 24</td>
<td>Introduction</td>
</tr>
<tr>
<td>January 26</td>
<td>Simple Program, Arithmetic</td>
</tr>
<tr>
<td>January 31</td>
<td>Memory Concepts</td>
</tr>
<tr>
<td>February 2</td>
<td>Algorithm, Flowchart, Pseudocode</td>
</tr>
<tr>
<td>February 7</td>
<td>Control Structure</td>
</tr>
<tr>
<td>February 9</td>
<td>Program Control, Iteration</td>
</tr>
<tr>
<td>February 14</td>
<td>Switch Case, Logical Operators</td>
</tr>
<tr>
<td>February 17</td>
<td>Functions</td>
</tr>
<tr>
<td>February 21</td>
<td>Recursions</td>
</tr>
<tr>
<td>February 23</td>
<td><strong>Midterm 1</strong></td>
</tr>
<tr>
<td>February 28</td>
<td>Arrays, Multidimensional Arrays</td>
</tr>
<tr>
<td>March 2</td>
<td>Searching, Sorting</td>
</tr>
<tr>
<td>March 7</td>
<td>Pointers</td>
</tr>
<tr>
<td>March 9</td>
<td>Array of pointers</td>
</tr>
<tr>
<td>March 14</td>
<td><strong>No classes, Spring Break</strong></td>
</tr>
<tr>
<td>March 16</td>
<td><strong>No classes, Spring Break</strong></td>
</tr>
<tr>
<td>March 21</td>
<td>Characters and Strings</td>
</tr>
<tr>
<td>March 23</td>
<td>Formatted Input/Output</td>
</tr>
<tr>
<td>March 28</td>
<td>Structures and Unions</td>
</tr>
<tr>
<td>April 4</td>
<td>Bit Manipulation, Enumeration</td>
</tr>
<tr>
<td>April 6</td>
<td><strong>Midterm 2</strong></td>
</tr>
<tr>
<td>April 11</td>
<td>No classes, Passover</td>
</tr>
<tr>
<td>April 13</td>
<td>File Processing</td>
</tr>
<tr>
<td>April 18</td>
<td>Programming embedded system – overview</td>
</tr>
<tr>
<td>April 20</td>
<td>Programming embedded system – i/o</td>
</tr>
<tr>
<td>April 25</td>
<td>Programming embedded system – condition</td>
</tr>
<tr>
<td>April 27</td>
<td>Programming embedded system – loop</td>
</tr>
<tr>
<td>May 2</td>
<td>Data Structure</td>
</tr>
<tr>
<td>May 4</td>
<td>Object Oriented Programming</td>
</tr>
<tr>
<td>May 9</td>
<td>Review</td>
</tr>
<tr>
<td>May 18</td>
<td>Final Exam 8:00-10:00AM</td>
</tr>
</tbody>
</table>
Practical use of Computers & Programming

- Electronic Health Records
- Human Genome Project
- AMBER Alert
- World Community Grid
- Medical Imaging
- One laptop per child
- Cloud Computing
- GPS
- Robots
- Email, Social Network
- Internet TV
- Game Programming
Computers: Hardware & Software

In use today are

- more than a billion general-purpose computers, and
- billions more *embedded* computers are used in cell phones, smartphones, tablet computers, home appliances, automobiles and more.

Computers can perform computations and make logical decisions phenomenally faster than human beings can
Terminology

- Computers process data under the control of sets of instructions called computer programs.
- These programs guide the computer through ordered actions specified by people called computer programmers.
- The programs that run on a computer are referred to as software.
Moore’s Law

- The number of transistors in a dense integrated circuit doubles approximately every two years.
Computer Organization

- Input Unit
- Output Unit
- Memory Unit
- Arithmetic & Logic Unit
- Central Processing Unit
- Secondary Storage Unit
Programming Languages

High Level Language

Assembly Language

Assembler

Machine Language

Compiler

Hardware

Interpreter
The C Language

- Currently, the most commonly-used language for embedded systems
- Very portable: compilers exist for virtually every processor
- Easy-to-understand compilation
- Produces efficient code
- Fairly concise
C History

- Developed between 1969 and 1973 along with Unix
- Due mostly to Dennis Ritchie
- Designed for systems programming
  - Operating systems
  - Utility programs
  - Compilers
  - Filters
- Evolved from B, which evolved from BCPL
C History

- Original machine (DEC PDP-11) was very small
  - 24K bytes of memory, 12K used for operating system

- Written when computers were big, capital equipment
  - Group would get one, develop new language, OS
C – Built for Performance

- Operating Systems
  - Unix, Linux, Android, portions of Windows, OS-X built on Objective C
- Embedded Systems
  - GPS, Intelligent Traffic Alert, Robots
- Real-time Systems
  - Air traffic control, Industrial automation
- Communication Systems
- C-based programming languages
  - Objective C, Java, Visual C#
C Program Development

Phase 1: Programmer creates program in the editor and stores it on disk.

Phase 2: Preprocessor program processes the code.

Phase 3: Compiler creates object code and stores it on disk.

Phase 4: Linker links the object code with the libraries, creates an executable file and stores it on disk.

Fig. 1.7 | Typical C development environment. (Part 1 of 3.)
Fig. 1.7 | Typical C development environment. (Part 2 of 3.)
**Phase 6:**
CPU takes each instruction and executes it, possibly storing new data values as the program executes.

**Fig. 1.7** | Typical C development environment. (Part 3 of 3.)
Environment to be used in class

- Cloud 9
  - [https://c9.io](https://c9.io)
  - Invitation will be sent to your albany.edu email to join