Introduction to Programming Concepts for Engineers

ENEE 140

Prof. Tudor Dumitraș
Assistant Professor, ECE
University of Maryland, College Park

http://ter.ps/enee140

Introducing the Teaching Staff

Instructor

Tudor Dumitraș

Office: AVW 3425

Email: tdumitra@umiacs.umd.edu

Course Website: http://ter.ps/enee140

Office Hours: Tue 2-3 pm

Undergraduate Teaching Fellows (with office hours)

Chris DeFrancisci

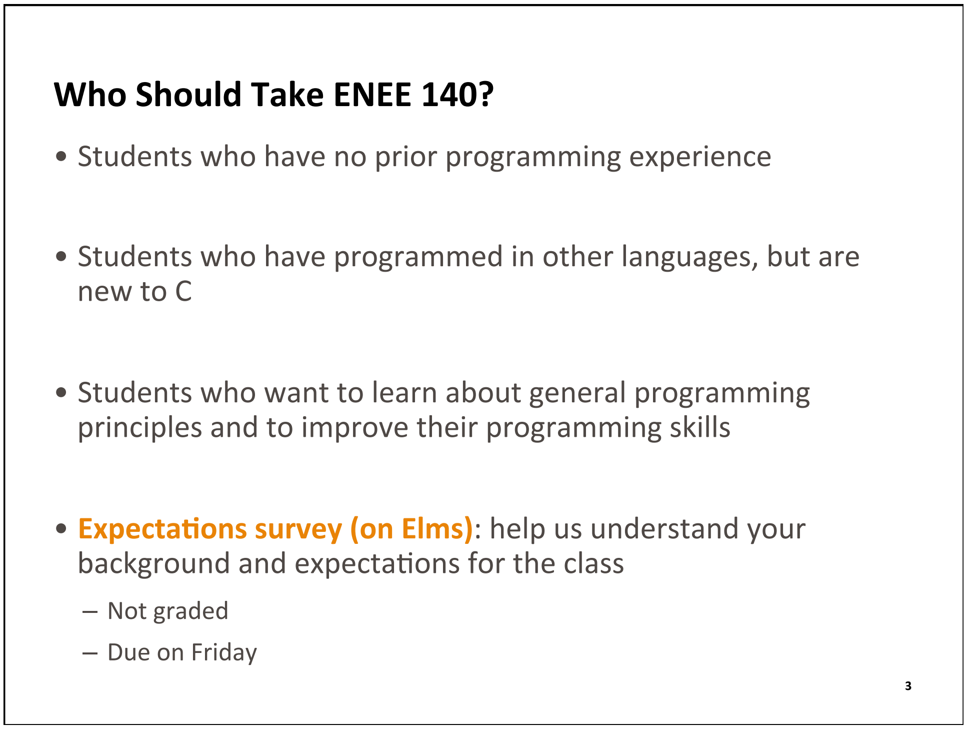
Tue 5-6 pm
AVW 1442

Juan Alonso

Wed 2:30-3:30 pm
AVW 1442

Eric Huang

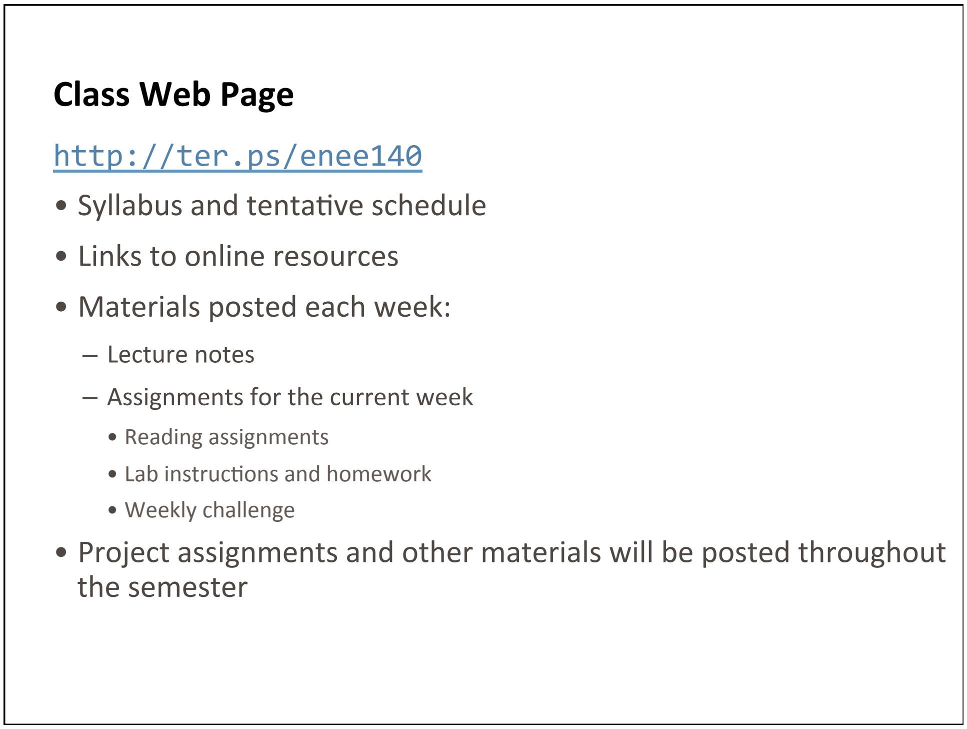
Thu 1-2 pm
AVW 1442



Who Should Take ENEE 140?

- Students who have no prior programming experience
- Students who have programmed in other languages, but are new to C
- Students who want to learn about general programming principles and to improve their programming skills
- **Expectations survey (on Elms)**: help us understand your background and expectations for the class
  - Not graded
  - Due on Friday

3



Class Web Page

http://ter.ps/enee140

- Syllabus and tentative schedule
- Links to online resources
- Materials posted each week:
  - Lecture notes
  - Assignments for the current week
    - Reading assignments
    - Lab instructions and homework
    - Weekly challenge
- Project assignments and other materials will be posted throughout the semester

Other Class Resources

`http://elms.umd.edu/`

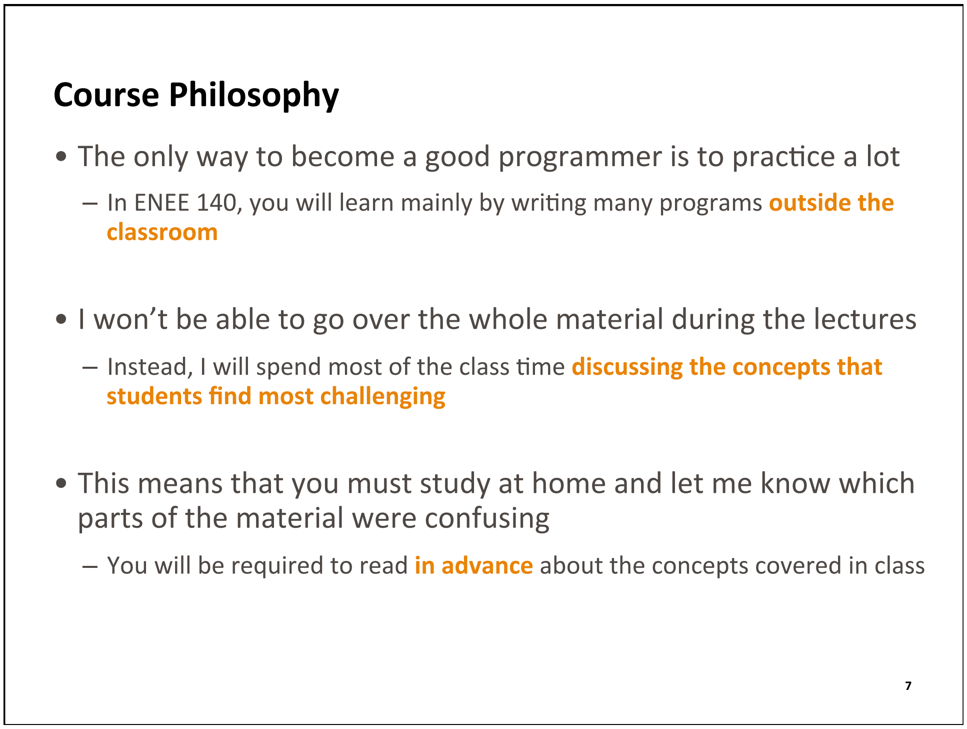
- **Grades**: view all scores (quizzes, homeworks, projects, exams)
- **Quizzes**: take quizzes and surveys online

Piazza

- Message forum for students and instructors
- Sign up at https://piazza.com/umd/spring2016/enee140/home
- Use the message board to ask questions and to receive announcements from the instructors

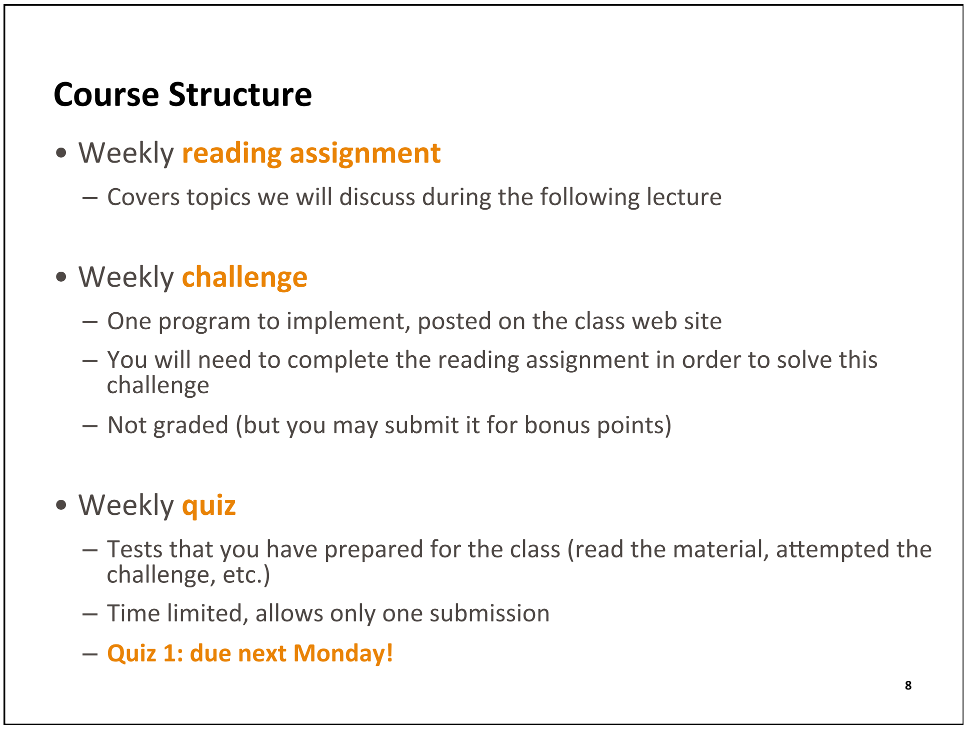
Tools We Will Use

- An Integrated Development Environment (IDE)
  - Should provide integrated access to all the tools needed for programming (e.g. code editor, project manager, compiler, debugger)
  - **I recommend you to install CLion** (https://www.jetbrains.com/clion/) on your laptops
  - Alternatively, you may use Eclipse, which is installed on the lab machines
- GRACE
  - Computing and file sharing environment
  - You will submit all your programming assignments using the submit command on the GRACE machines
  - You can launch Eclipse remotely from the GRACE lab machines (see the handout on "How to Launch Eclipse")
  - Class directory: `/afs/glue.umd.edu/class/spring2016/enee/140/0101`
    - Code examples posted in the `public/` directory



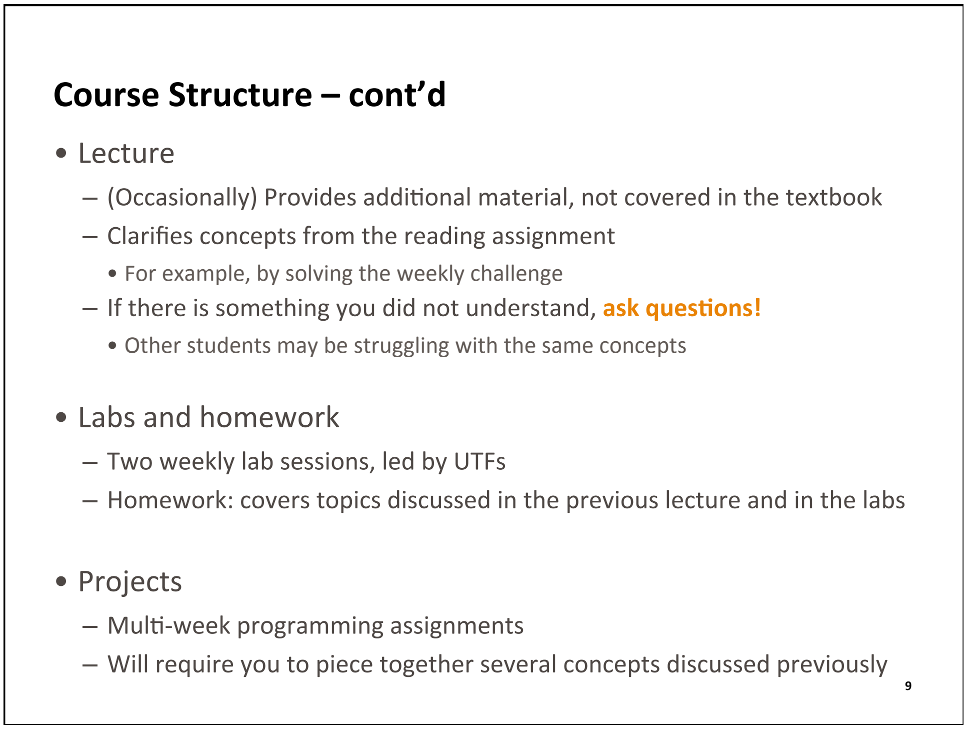
Course Philosophy

- The only way to become a good programmer is to practice a lot
  - In ENEE 140, you will learn mainly by writing many programs **outside the classroom**
- I won't be able to go over the whole material during the lectures
  - Instead, I will spend most of the class time **discussing the concepts that students find most challenging**
- This means that you must study at home and let me know which parts of the material were confusing
  - You will be required to read **in advance** about the concepts covered in class



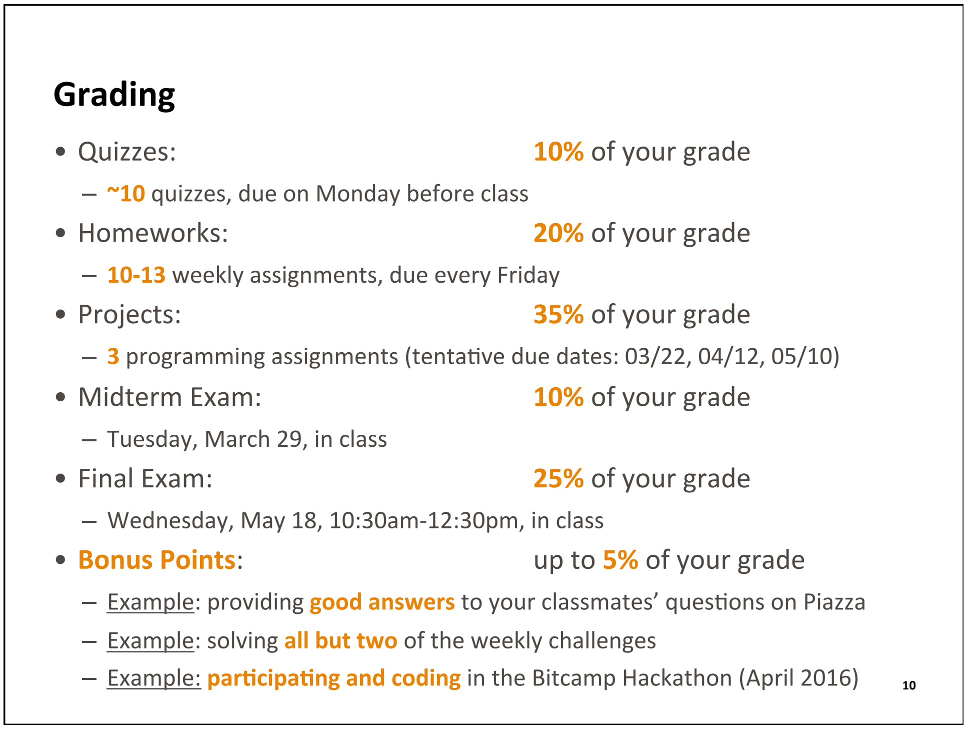
Course Structure

- Weekly **reading assignment**
  - Covers topics we will discuss during the following lecture
- Weekly **challenge**
  - One program to implement, posted on the class web site
  - You will need to complete the reading assignment in order to solve this challenge
  - Not graded (but you may submit it for bonus points)
- Weekly **quiz**
  - Tests that you have prepared for the class (read the material, attempted the challenge, etc.)
  - Time limited, allows only one submission
  - **Quiz 1: due next Monday!**



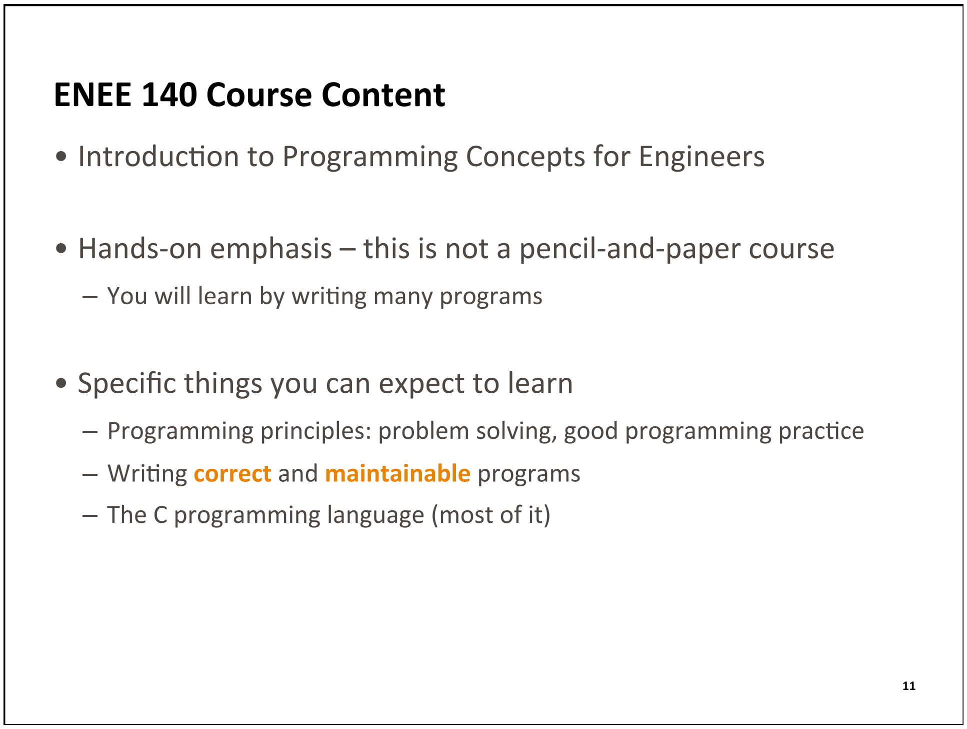
Course Structure – cont’d

- Lecture
  - (Occasionally) Provides additional material, not covered in the textbook
  - Clarifies concepts from the reading assignment
    - For example, by solving the weekly challenge
  - If there is something you did not understand, **ask questions!**
    - Other students may be struggling with the same concepts
- Labs and homework
  - Two weekly lab sessions, led by UTFs
  - Homework: covers topics discussed in the previous lecture and in the labs
- Projects
  - Multi-week programming assignments
  - Will require you to piece together several concepts discussed previously



Grading

- Quizzes: **10%** of your grade
  - **~10** quizzes, due on Monday before class
- Homeworks: **20%** of your grade
  - **10-13** weekly assignments, due every Friday
- Projects: **35%** of your grade
  - **3** programming assignments (tentative due dates: 03/22, 04/12, 05/10)
- Midterm Exam: **10%** of your grade
  - Tuesday, March 29, in class
- Final Exam: **25%** of your grade
  - Wednesday, May 18, 10:30am-12:30pm, in class
- **Bonus Points**: up to **5%** of your grade
  - Example: providing **good answers** to your classmates’ questions on Piazza
  - Example: solving **all but two** of the weekly challenges
  - Example: **participating and coding** in the Bitcamp Hackathon (April 2016)

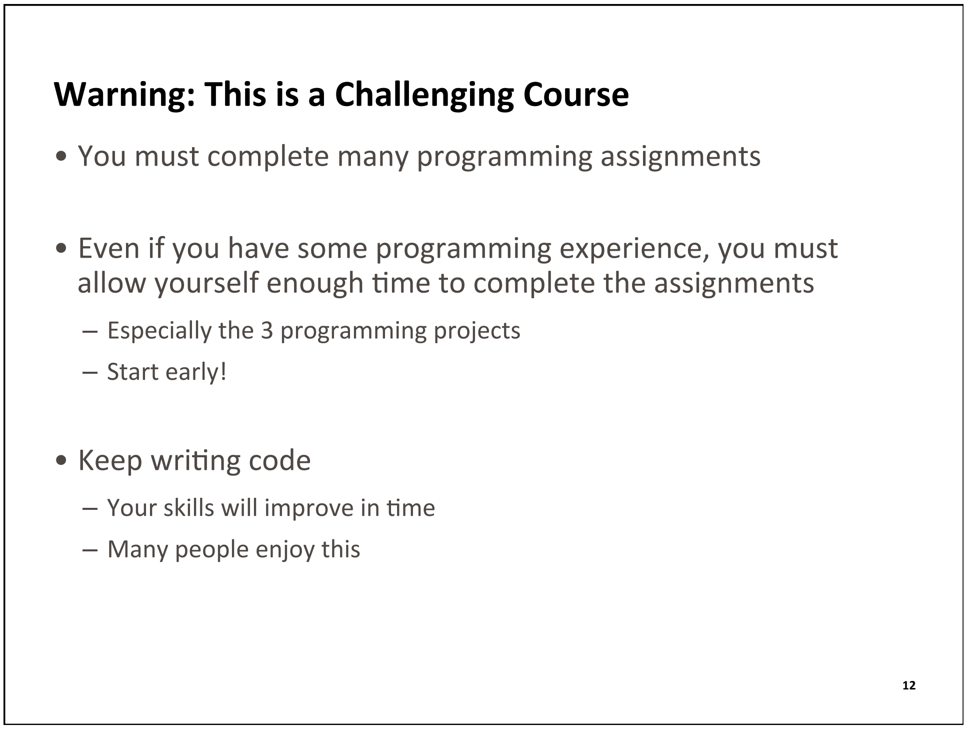


ENEE 140 Course Content

- Introduction to Programming Concepts for Engineers

- Hands-on emphasis – this is not a pencil-and-paper course
  - You will learn by writing many programs

- Specific things you can expect to learn
  - Programming principles: problem solving, good programming practice
  - Writing **correct** and **maintainable** programs
  - The C programming language (most of it)

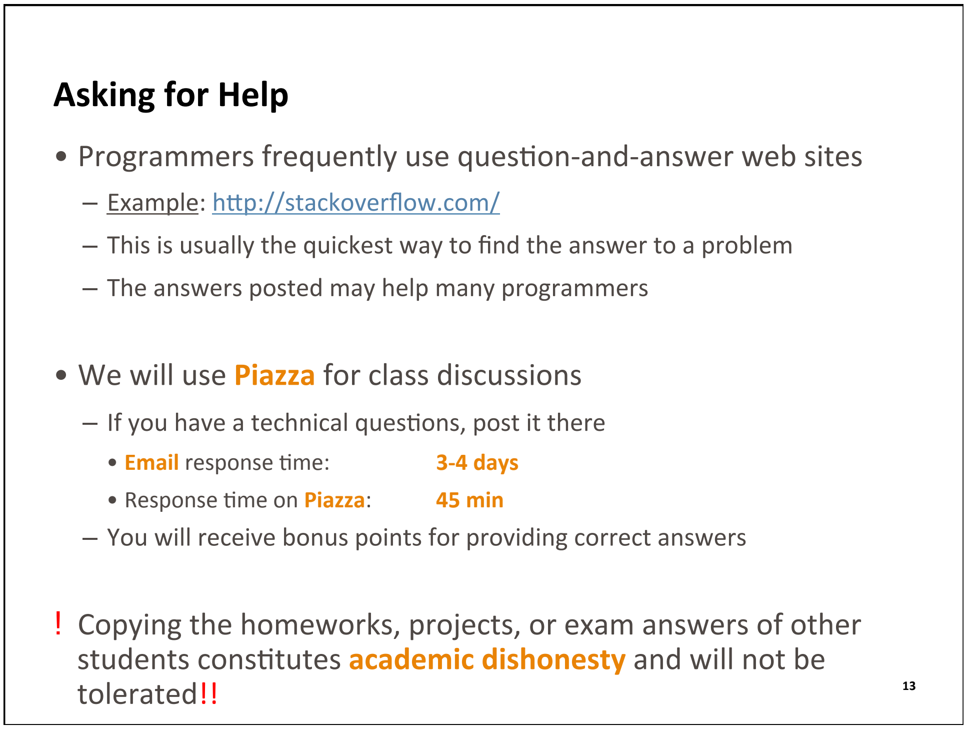


Warning: This is a Challenging Course

- You must complete many programming assignments

- Even if you have some programming experience, you must allow yourself enough time to complete the assignments
  - Especially the 3 programming projects
  - Start early!

- Keep writing code
  - Your skills will improve in time
  - Many people enjoy this

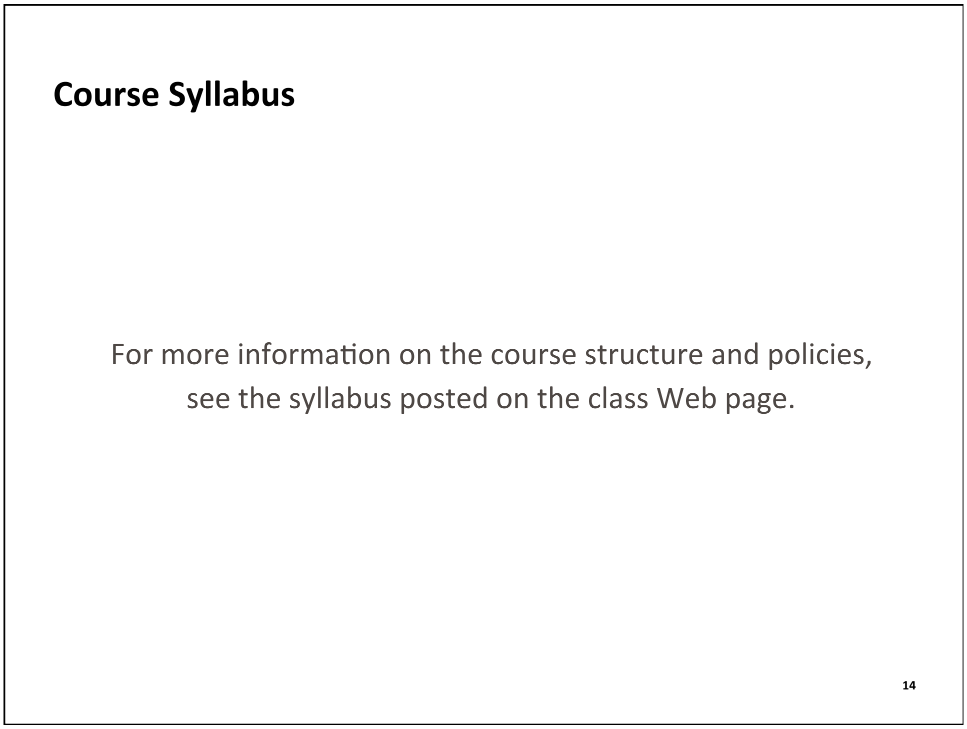


Asking for Help

- Programmers frequently use question-and-answer web sites
  - Example: http://stackoverflow.com/
  - This is usually the quickest way to find the answer to a problem
  - The answers posted may help many programmers

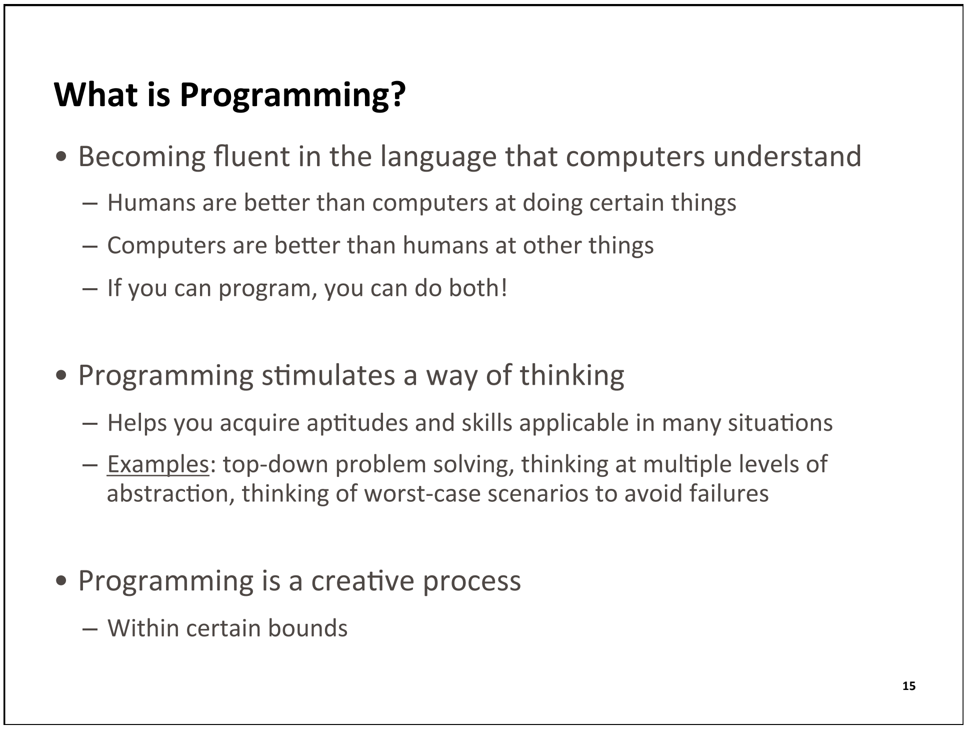
- We will use **Piazza** for class discussions
  - If you have a technical questions, post it there
    - **Email** response time: **3-4 days**
    - Response time on **Piazza**: **45 min**
  - You will receive bonus points for providing correct answers

! Copying the homeworks, projects, or exam answers of other students constitutes **academic dishonesty** and will not be tolerated!!



Course Syllabus

For more information on the course structure and policies,
see the syllabus posted on the class Web page.



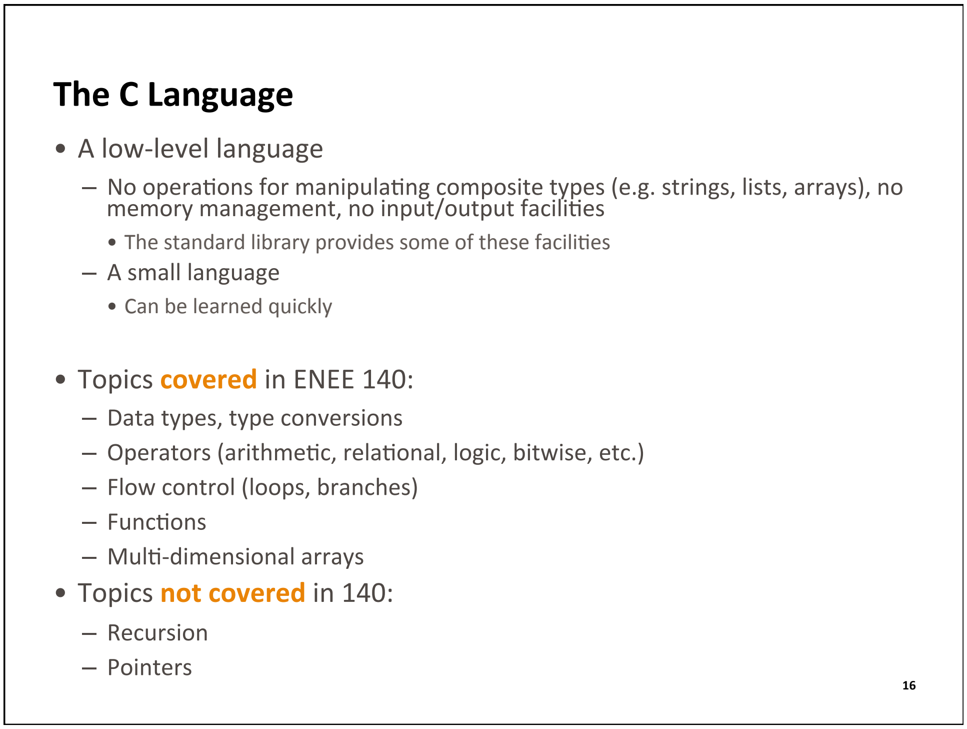
What is Programming?

- Becoming fluent in the language that computers understand
  - Humans are better than computers at doing certain things
  - Computers are better than humans at other things
  - If you can program, you can do both!

- Programming stimulates a way of thinking
  - Helps you acquire aptitudes and skills applicable in many situations
  - Examples: top-down problem solving, thinking at multiple levels of abstraction, thinking of worst-case scenarios to avoid failures

- Programming is a creative process
  - Within certain bounds

15



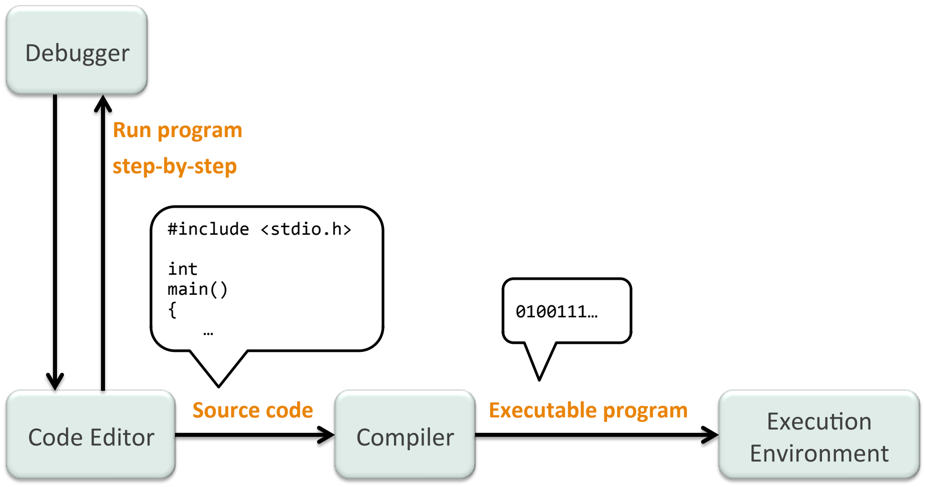
The C Language

- A low-level language
  - No operations for manipulating composite types (e.g. strings, lists, arrays), no memory management, no input/output facilities
    - The standard library provides some of these facilities
  - A small language
    - Can be learned quickly

- Topics **covered** in ENEE 140:
  - Data types, type conversions
  - Operators (arithmetic, relational, logic, bitwise, etc.)
  - Flow control (loops, branches)
  - Functions
  - Multi-dimensional arrays
- Topics **not covered** in 140:
  - Recursion
  - Pointers

16

The Programming Toolchain



```
#include <stdio.h>

int
main()
{
    …
```

The Textbook

- B. Kernighan and D. Ritchie, *The C Programming Language*, 2nd edition, Prentice Hall 1988.
  - Also known as K&R
  - Every serious programmer should have this book in his/her library
  - Not an easy book for beginners (assumes prior programming knowledge)

- If you have no programming experience
  - Read the chapters assigned each week in the lecture
  - Consult Steve Summit's excellent notes on the textbook: http://www.eskimo.com/~scs/cclass/krnotes/top.html
  - I will try to clarify the most important points in the lecture

- For this week: read the **Introduction and Chapters 1.1, 1.2, 1.4**

ENEE 140 Focuses on Programming Principles

- The lectures will discuss important programming principles
  - Most of these are applicable to any programming language
  - C examples will be provided for illustration

- To learn all the details about the C concepts discussed, you must read additional materials
  - The relevant chapters in the textbook
  - Many Internet resources on C programming (Google is your friend)
  - Quick documentation: press F1 in CLion or move your mouse over a function in Eclipse

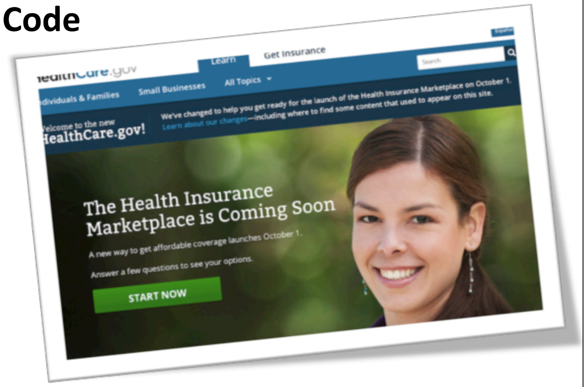
First Principles: Code Quality

- Learning objective: write high-quality code
  - **Correctness**: the code should do what it's supposed to do (and nothing else!)
  - **Maintainability**: other programmers should find the code easy to read and to modify

- Other quality attributes that we will not emphasize in ENEE 140
  - Efficiency
  - Robustness
  - Security

Example of Incorrect Code

- Healthcare.gov
  - Launched on October 1st 2013
  - Only 1% of interested people managed to enroll in the first week of operation



elcome to the new HealthCare.gov!

We've changed to help you get ready for the launch of the Health Insurance Marketplace on October 1. Learn about our changes—including where to find some content that used to appear on this site.

The Health Insurance Marketplace is Coming Soon

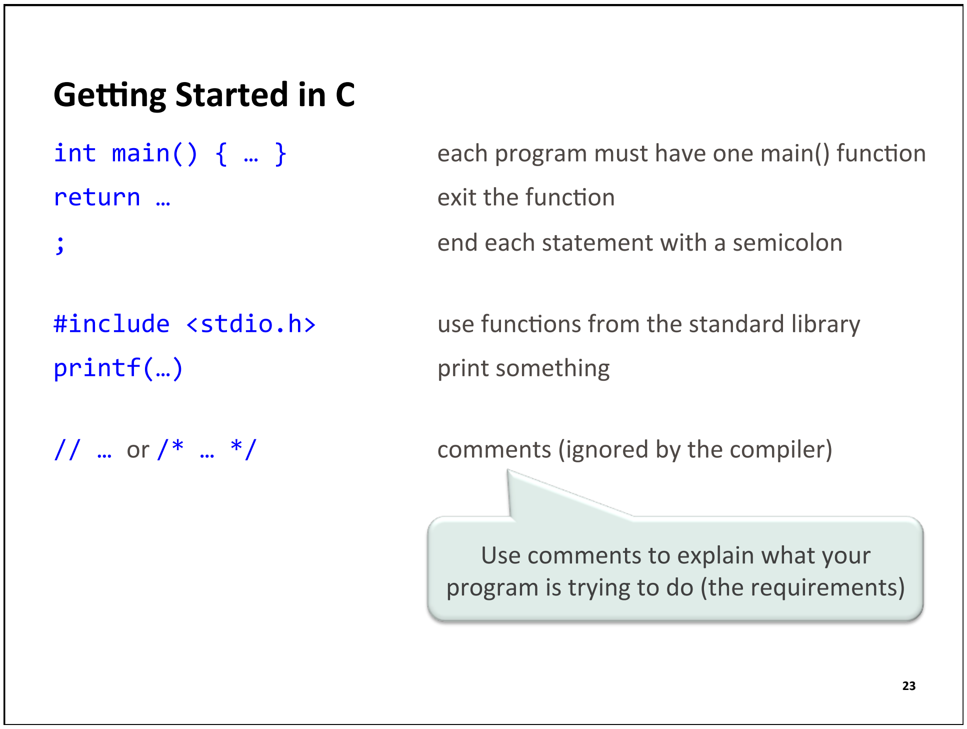
A new way to get affordable coverage launches October 1.

Answer a few questions to see your options.

  - The software was not designed for the 100,000s of concurrent users it attracted after launch
  - States continued adding new rules and clarifications after October 1st

Requirements

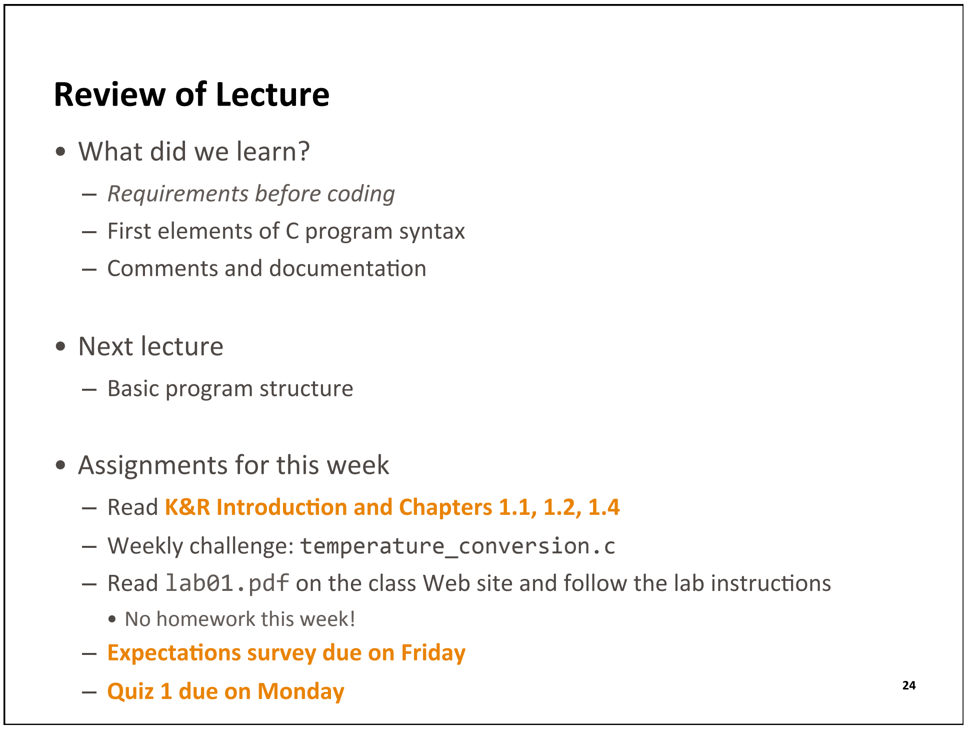
Before you start programming,
you must understand the requirements
(you must **know what the program is supposed to do**)



Getting Started in C

`int main() { … }`	each program must have one main() function
`return …`	exit the function
`;`	end each statement with a semicolon
`#include <stdio.h>`	use functions from the standard library
`printf(…)`	print something
`// …` or `/* … */`	comments (ignored by the compiler)

Use comments to explain what your program is trying to do (the requirements)



Review of Lecture

- What did we learn?
  - *Requirements before coding*
  - First elements of C program syntax
  - Comments and documentation
- Next lecture
  - Basic program structure
- Assignments for this week
  - Read **K&R Introduction and Chapters 1.1, 1.2, 1.4**
  - Weekly challenge: `temperature_conversion.c`
  - Read `lab01.pdf` on the class Web site and follow the lab instructions
    - No homework this week!
  - **Expectations survey due on Friday**
  - **Quiz 1 due on Monday**