

Introduction to Programming Concepts for Engineers

ENEE 140

Prof. Tudor Dumitras

Assistant Professor, ECE
University of Maryland, College Park



<http://ter.ps/enee140>

Introducing Your Instructor



Tudor Dumitras

Office: AVW 3425

Email: tdumitra@umiacs.umd.edu

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

Office Hours: Tue 2-3 pm

Class Resources

<http://e1ms.umd.edu/>

- **Modules:** lab and lecture materials
- **Assignments:** information about projects, lab reports and exams
- **My grades:** all scores (lab reports, projects, exams)
- **Piazza:** message board for students and instructors
 - Sign up at <http://piazza.com/umd/spring2015/enee140>
 - Use the message board to ask questions and to receive announcements from the instructors
- The following materials will be posted each week:
 - Lecture notes
 - Lab materials for the current week
 - Lab instructions
 - Homework
 - Challenge for next week
- Project assignments and other materials will be posted throughout the semester

Tools We Will Use

- Eclipse
 - Integrated Development Environment (IDE) for C and other programming languages
 - Provides integrated access to all the tools needed for programming (e.g. code editor, project manager, compiler, debugger)
- GRACE
 - Computer 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/spring2015/enee/140/0101`
 - Code examples posted in the `public/` directory

4

Course Structure

- Weekly **reading assignment**
 - Covers topics we will discuss during the following lecture
- Weekly **challenge**
 - One program to implement, posted online
 - You will need to complete the reading assignment in order to solve this challenge
 - Not graded
- 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

5

Grading

- Homeworks: **20%** of your grade
 - **10-13** weekly recitation assignments, due every Friday.
 - Separate from the weekly challenges
- Projects: **40%** of your grade
 - **3** programming assignments
 - Tentative due dates: March 23, April 13, May 11
- Midterm Exam: **15%** of your grade
 - Tuesday, March 31 in class
- Final Exam: **25%** of your grade
 - Tuesday, Wednesday, May 20, 10:30am-12:30pm, in class
- **Bonus Points:** up to **10%** of your grade
 - Example: answering your classmates' questions in the **Discussions** section of the web site
 - Example: volunteering to help with a Maryland Day activity
 - Example: participate in the Bitcamp Hackathon (April 10-12)

6

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)

7

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

8

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
 - If you ask me a technical question during office hours, I will first request you to summarize the replies to your question on the message board
 - You will receive bonus points for answering questions on the message board
- ! Copying the homeworks, projects, or exam answers of other students constitutes **academic dishonesty** and will not be tolerated!!

9

Course Syllabus

For more information on the course structure and policies,
see the syllabus posted on Elms.

10

What is Programming?

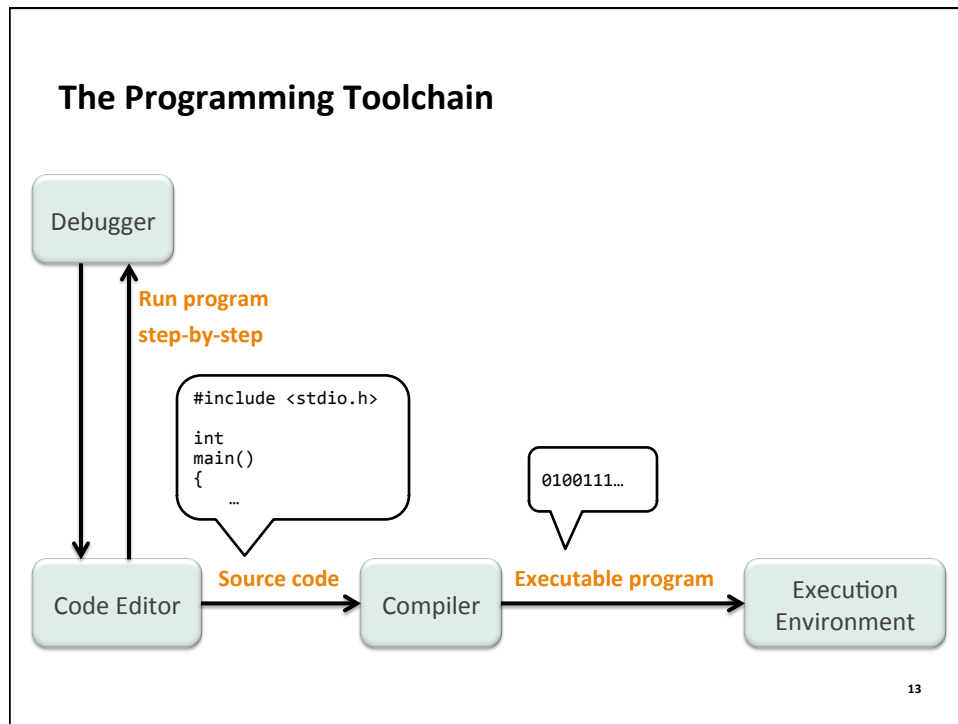
- Becoming fluent in the language that computers understand
 - Humans are better than computers at doing certain things
 - Computers are better than humans at other things
 - If you can program, you can do both!
- Programming stimulates a way of thinking
 - Helps you acquire aptitudes and skills applicable 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

11

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

12



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 can program in another language (e.g. Java, Python)
 - Read Chapter 1 for a C tutorial
 - If you have no programming experience
 - Don't read Chapter 1 at first (it will confuse you)
 - Instead, read the chapters suggested each week in the lecture
 - I will try to clarify the most important points in the lecture notes
 - See also Steve Summit's excellent notes on the textbook: <http://www.eskimo.com/~scs/c/class/krnotes/top.html>
 - For this week: read **Chapters 1.1, 1.2 and 1.4**
- 14

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)
 - Eclipse hints (move your mouse over a function)

15

First Principles: Code Quality

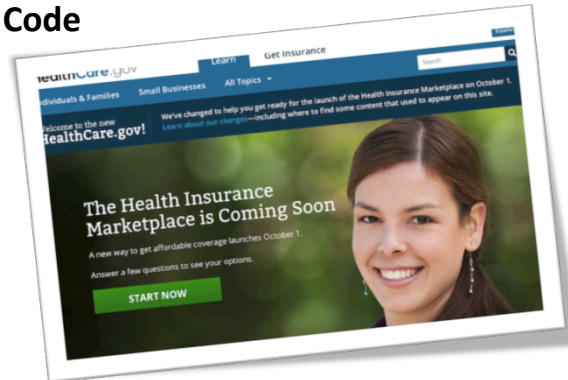
- Learning objective: write high-quality code
 - **Correctness**: the code should do what it's supposed to do (and nothing else!)
 - **Maintainability**: other programmers should find the code easy to read and to modify
- Other quality attributes that we will not emphasize in ENEE 140
 - Efficiency
 - Robustness
 - Security

16

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
- 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



17

Requirements

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

18

Getting Started in C

```
int main() { ... }
return ...
;
```

each program must have one main() function
exit the function
end each statement with a semicolon

```
#include <stdio.h>
printf(...)
```

use functions from the standard library
print something

```
// ... or /* ... */
```

comments (ignored by the compiler)

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

19

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 Chapters 1.1, 1.2 and 1.4**
 - Weekly challenge: `temperature_conversion.c`
 - Read `lab01.pdf` on Elms and follow the lab instructions
 - No homework this week!

20