ENEE 140 — Introduction to Programming Concepts for Engineers

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

http://ter.ps/enee140

1

Introducing the Teaching Staff

Instructor

Tudor Dumitraș
Email: tudor@umd.edu
Course Website: http://ter.ps/enee140

2

2

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 the bounds of what computers and programming languages can do

3

3

A Problem Solved by Programming

- Given a month and a year in the future, print the calendar on a text terminal
  - Example for October 2025

```
Su Mo Tu We Th Fr Sa
                1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

4

4

A Problem Solved by Programming

- Given a month and a year in the future, print the calendar on a text terminal
  - Example for October 2025

```
Su Mo Tu We Th Fr Sa
                1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

5

5

A Problem Solved by Programming

- Given a month and a year in the future, print the calendar on a text terminal
  - Example for October 2025

```
Su Mo Tu We Th Fr Sa
                1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

6

6

A Problem Solved by Programming

- Given a month and a year in the future, print the calendar on a text terminal
  - Example for October 2025

```
Su Mo Tu We Th Fr Sa
                1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

7

7

A Problem Solved by Programming

- Given a month and a year in the future, print the calendar on a text terminal
  - Example for October 2025

```
Su Mo Tu We Th Fr Sa
                1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

8

8

Review

- Write programs to compute quantities difficult to work out in your head
  - Programming languages provide **variables** and **arithmetic operations**
- Come up with step-by-step procedure for arriving at the result
  - Programming languages provide **loops** and **branching statements**
- Break down a problem into simpler steps
  - Programming languages provide **functions**
  - Helpful for solving problems from the top down to the small details
- Communicate with the user
  - Programming languages provide **input/output** mechanisms

Before writing a computer program, think about the program's design!

9

9

Course Philosophy

10

10

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

11

11

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

12

12

Course Structure: Before Each Lecture

- 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, just for practicing
    - **If you get it right: replace a homework problem of your choice**
- Weekly **quiz**
  - Tests that you have prepared for the class (read the material, attempted the challenge, etc.)
  - Time limited, allows only one submission

13

13

Course Structure: During/After Each Lecture

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

14

14

Class Resources

Class web page:

- **Lecture slides** posted before each lecture
- All assignments (**homeworks**, **challenges**, **projects**, links to **quizzes**)
- Links to online resources

Elms: http://elms.umd.edu/

- **Grades**: view all scores (quizzes, homeworks, projects, exams)
- **Quizzes**: take quizzes and surveys online
- **Gradescope**: submit all your programming assignments

Piazza

- **Discussion forum** for students and instructors
- Use the message board to ask questions and to receive announcements from the instructors
- Sign up link on class web page; access code:

15

15

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
  - CLion is also installed on the lab machines
- GRACE
  - Computing and file sharing environment
  - Materials posted under the class directory
    - Code examples posted under `public/code`
    - Lab instructions and homework under `public/labs`
    - Weekly challenge under `public/challenges`
    - Project code templates under `public/projects`

16

16

Grading

- Quizzes: **10%** of your grade
  - **~10** quizzes, due on Sunday before class
- Homeworks: **20%** of your grade
  - **10-13** weekly assignments, due every Friday
- Projects: **35%** of your grade
  - **2** programming assignments
- Midterm Exam: **10%** of your grade
- Final Exam: **25%** of your grade
- **Bonus Points**: up to **5%** of your grade
  - Example: providing **good answers** to your classmates' questions on Piazza

17

17

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)

18

18

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 (but see below)
- 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**

19

19

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 2 programming projects
  - Start early!
- Keep writing code
  - Your skills will improve in time
  - Many people enjoy this

20

20

Asking for Help

- In ENEE 140 we have an instructor for ~12 students
  - **Come see us during office hours!**
- Programmers frequently use question-and-answer web sites
  - Example: http://stackoverflow.com/
  - 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!!

21

21

Course Syllabus

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

The first quiz will test your understanding of the ENEE 140 syllabus

22

22

Review of Lecture

- What did we learn?
  - *Thinking like a programmer*
- 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!

23

23

Dive In

http://ter.ps/enee140

24

24