Ceng 111 – Fall 2015
Week 1b

Computation

Credit: Some slides are from the “Invitation to Computer Science” book by G. M. Schneider, J. L. Gersting and some from the “Digital Design” book by M. M. Mano and M. D. Ciletti.
Syllabus

CENG 111
Introduction to Computer Engineering Concepts
2015 Fall

Instructors:
Ahmet Coşar (section 1), Sinan Kalkan (section 2), Göktürk Üçoluk (section 3)

Teaching Assistants:
Akif Akkuş, Merve Asiler, Hüseyin Aydın, Alper Demir, Hazal Moğultay, Ahmet Rifaioğlu, Okan Tarhan Tursun

Course Schedule:

<table>
<thead>
<tr>
<th>Section 1</th>
<th>Section 2</th>
<th>Section 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tue: 13:40 (BMB1)</td>
<td>Mon: 12:40 (BMB4)</td>
<td>Mon: 9:40 (BMB3)</td>
</tr>
</tbody>
</table>
Labs

- There will be labs almost every week.
- Labs are complementary to the lectures
  - Lectures are mostly theoretical
  - Labs give you the practical knowledge/abilities
- Attendance is mandatory
- Take them extremely seriously!
Content

Catalog Description:
Introduction to the fundamentals of computer systems, including computer organization, operating systems, language processors and user interfaces. Introduction to algorithms and programming. Reasoning informally about the correctness and efficiency of programs. A functional programming language will be used for practical work.

1/3 of the course:
- What is CS, computing, computation, algorithm?
- Inside of a Computer: CPU, Memory, Gates, Boolean Logic.
- Von Neumann architecture
- Etc.

2/3 of the course:
- Programming concepts with Python.
Textbooks

- For 1.part: “Invitation to Computer Science”, by G. Schneider and J. Gersting, West.
  - Downloadable from:

  - Downloadable in METU, for URL see Homepage>>Library

- Both books can be obtained in print from the photocopy shop at the department.
- Students must preregister in signing to the shop. (the price is the photocopying cost only)
http://www.ceng.metu.edu.tr/course/ceng111
CENG account

user name/passwd has been mailed to your METU account (the one used for registration).

- use

webmail.metu.edu.tr

to view mails there.
Grading:

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>18%</td>
</tr>
<tr>
<td>Take-home Exams (4)</td>
<td>25%</td>
</tr>
<tr>
<td>Labs</td>
<td>32%</td>
</tr>
<tr>
<td>Participation</td>
<td>3%</td>
</tr>
<tr>
<td>Final</td>
<td>22%</td>
</tr>
</tbody>
</table>

- Most assignments will be programming assignments (some could be exercises to be written up).
- Laboratory schedule will be announced later. Keep watching.
- Starting from week 4-5 the laboratory sessions will go in parallel with the lectures. You will be asked to solve problems by using the ideas introduced in the lectures of that week. Earlier lab sessions will build your computing/communicating literacy.
Take-home Exams

We will have four take-home exams:

- Essentially programming homeworks but for official reasons, we name them “take-home exams”

They are extremely important:

- You get the chance to practice what you learn
- You can get your hands “dirty”
- Well, they are your first “few walking steps” towards programming
In this department, we teach you a new occupation!

- In this sense, if you cheat, you fool yourselves; not us!

- Cheating is not acceptable in this department or any other departments or teaching institutions.

- Rent-a-coder

- Asking help in forums

- Violation => disciplinary action!!
Today

- What is computer science?
- Computation, Computing
- Digital Computation
WHAT IS COMPUTER SCIENCE?
What is Computer Science?

- COMMON MISCONCEPTIONS (1)
  - “Computer science is the study of computers”

- Incomplete – theoretical work began (1920-1940) before computers
- CS became an independent field of study late 1950’s, early 1960’s
- Theoretical CS – relies on formal models rather than “real” machines
- CS “is no more about computers than astronomy is about telescopes”, etc.

From “Invitation to Computer Science”
What is Computer Science?

- COMMON MISCONCEPTIONS (2)
  - “Computer science is the study of how to write computer programs”
  - Programming is important, but it is just a tool for studying new ideas, representing information or testing the solution to a problem.
  - A Program is a means to an end, not the end itself.
  - E.g. Searching a list such as the NYC phone Directory

From “Invitation to Computer Science”
What is Computer Science?

**COMMON MISCONCEPTIONS (3)**

- “Computer science is the study of the uses and applications of computers and software”

- E.g. word processors, databases, spreadsheets, etc.

- Many people USE software, but the Computer Scientist is responsible for specifying, designing, building and testing software packages and the systems on which they run.

From “Invitation to Computer Science”
What is Computer Science?

All of the following concepts are incomplete and do not capture the richness and diversity of this exciting field:

- computers,
- programming languages,
- software applications, and uses.
Computer Science is the study of algorithms (= methods)

including:

1. their formal and mathematical properties
2. their hardware realizations
3. their linguistic realizations
4. their applications

From “Invitation to Computer Science”
Computer Science is the study of Algorithms

Including:

1. their formal and mathematical properties
   - studying the behavior of algorithms to see that they are correct and efficient

From “Invitation to Computer Science”
Computer Science is the study of Algorithms

Including:

2. their hardware realizations
   - designing and building computer systems to execute the algorithms

From “Invitation to Computer Science”
Computer Science is the study of Algorithms

Including:

3. their linguistic realizations
   • designing programming languages and translating the algorithms into these languages so that they can be executed by the hardware

From “Invitation to Computer Science”
Computer Science is the study of Algorithms

Including:

4. their applications

- identifying important problems and designing correct and efficient software packages to solve them.

From “Invitation to Computer Science”
Measure the height of a tall building with a barometer

What would be your answer?

One student answered:

- "I would tie the barometer to a rope, hang it down from the top of the building to the bottom and measure the length of the rope!"

- Of course, the instructor rejects the answer since it doesn’t include any "physics"

Check the following for two different versions of the `legend': http://www.snopes.com/college/exam/barometer.asp
Where does the word ‘algorithm’ come from?

From a Persian mathematician, astronomer and geographer: Mohammed ibn-Musa al-Khwarizmi

“Algorithmi” is the latin form of his name

He contributed to science by

- Decimal positional number system
  (e.g., $32 = 10^1 \times 3 + 10^0 \times 2$)
- Presented the first systematic solutions to linear and quadratic equations

In fact, the word “Algebra” comes from one of his operators (al-jabr: subtracting a number from both sides of an equation) for solving equations

Where does the word ‘algorithm’ come from? (cont’d)

al-Khwarizmi reduced equations to one of the following six forms by using al-jabr (in Arabic: restoring, completion):

- squares equal roots \((ax^2 = bx)\)
- squares equal number \((ax^2 = c)\)
- roots equal number \((bx = c)\)
- squares and roots equal number \((ax^2 + bx = c)\)
- squares and number equal roots \((ax^2 + c = bx)\)
- roots and number equal squares \((bx + c = ax^2)\)

For example, \(x^2 = 40x - 4x^2\) is reduced to \(5x^2 = 40x\). From this reduced form, it is easily deducable that the variable is either 0 or 8.

What does ‘algorithm’ mean?

- “A procedure or formula for solving a problem”
- “A set of instructions to be followed to solve a problem”
- “an effective method expressed as a finite list of well-defined instructions for calculating a function”
- “step-by-step procedure for calculations”
What is an algorithm?

- An algorithm is a list that looks like
  - STEP 1: Do something
  - STEP 2: Do something
  - STEP 3: Do something
  - ... ...
  - STEP N: Stop, you are finished

From “Invitation to Computer Science”
"I think you should be more explicit here in step two."

From “Invitation to Computer Science”
Valid Operations in Algorithms

- **Sequential** – simple well-defined task, usually declarative sentence.
- **Conditional** - “ask a question and select the next operation on the basis of the answer to the question – usually an “if-then” or “if then else”
- **Iterative** - “looping” instructions – repeat a set of instructions

From “Invitation to Computer Science”
Algorithms

- We use them all the time.
- Can you give examples?
  - Following directions
  - Recording a DVD
  - Adding two numbers
  - Finding Greatest Common Divisor
  - ...

From “Invitation to Computer Science”
An example algorithm

Algorithm for Adding Two $m$-Digit Numbers

Given: $m \geq 1$ and two positive numbers each containing $m$ digits, $a_{m-1} a_{m-2} \ldots a_0$ and $b_{m-1} b_{m-2} \ldots b_0$

Wanted: $c_m c_{m-1} c_{m-2} \ldots c_0$, where $c_m c_{m-1} c_{m-2} \ldots c_0 = (a_{m-1} a_{m-2} \ldots a_0) + (b_{m-1} b_{m-2} \ldots b_0)$

Algorithm:

Step 1  Set the value of carry to 0.
Step 2  Set the value of $i$ to 0.
Step 3  While the value of $i$ is less than or equal to $m - 1$, repeat the instructions in steps 4 through 6.
Step 4  Add the two digits $a_i$ and $b_i$ to the current value of carry to get $c_i$.
Step 5  If $c_i \geq 10$, then reset $c_i$ to $(c_i - 10)$ and reset the value of carry to 1; otherwise, set the new value of carry to 0.
Step 6  Add 1 to $i$, effectively moving one column to the left.
Step 7  Set $c_m$ to the value of carry.
Step 8  Print out the final answer, $c_m c_{m-1} c_{m-2} \ldots c_0$.
Step 9  Stop.

From “Invitation to Computer Science”
Why are algorithms important?

- If we can specify an algorithm to solve a problem then we can automate its solution.

- No algorithm => No software => No automation!

From “Invitation to Computer Science”
Can we find algorithms to all problems?

NO!

- There are problems which have no generalized solutions – unsolvable or intractable
- Some with an algorithm would take so long to execute that the algorithm is useless
- Some problems we have not yet discovered an algorithm for

From “Invitation to Computer Science”
A formal definition of algorithm

“Starting from an initial state and initial input (perhaps empty), the instructions describe a computation that, when executed, will proceed through a finite number of well-defined successive states, eventually producing "output" and terminating at a final ending state.”
“Computation”

- Digital vs. analog computation
- Sequential vs. parallel computation
- Batch vs. interactive computation
- Evolutionary, molecular, quantum computation

“Physical computation” / “Digital Physics”
- ‘The whole universe is itself a computation’
Problem: Find temperature of the water if A&B were mixed together.

Any suggestions on how to solve it?
Computation in our brain

- Highly-connected network of neurons.
- How many neurons?
  - Approx. $10^{11}$ neurons and $10^{14}$ synapses.
- How do they transmit information?
  - Using nothing else than charged molecules.
Computation in our brain (cont’d)

- Each neuron gets input and produces an output using an “activation function”
Some of ours’ is smaller but they have essentially the same computational mechanisms! 😊
DIGITAL COMPUTATION
BUT FIRST SOME HISTORICAL OVERVIEW
The Early Period: Up to 1940

- 3,000 years ago: Mathematics, logic, and numerical computation
  - Important contributions made by the Greeks, Egyptians, Babylonians, Indians, Chinese, and Persians
  - Cuneiform
  - Stone “abacus”

- [http://www.thocp.net/slideshow/0469.htm](http://www.thocp.net/slideshow/0469.htm)
ABACUS

Early calculating devices

ABACUS – 2700 BC (Mesopotamia)

http://www.computersciencelab.com/ComputerHistory/History.htm

Slide from “Introduction to Computing”
DaVinci

- 1452-1519 Leonardo DaVinci sketched gear-driven calculating machines but none were ever built.

http://www.computersciencelab.com/ComputerHistory/History.htm
Napier’s Bones

1614: Logarithms

- Invented by John Napier to simplify difficult mathematical computations

http://www.computersciencelab.com/ComputerHistory/History.htm
If you want to multiply 7 by 46785499:
Slide Rule (slipstick)  
“a mechanical analog computer” 

Around 1622: First slide rule created

http://www.computersciencelab.com/ComputerHistory/History.htm
The Pascaline: One of the Earliest Mechanical Calculators
The Early Period: Up to 1940

Jacquard’s Loom

Also see http://www.computersciencelab.com/ComputerHistory/HistoryPt2.htm
Difference engine

http://www.youtube.com/watch?v=0anIyVGeWOI
The Harvard Mark-I

Grace M. Hopper working on the Harvard Mark-I, developed by IBM and Howard Aiken. The Mark-I remained in use at Harvard until 1959, even though other machines had surpassed it in performance, providing vital calculations for the navy in World War II.
Programming the ENIAC
History of Computation

- Read the reading material on this subject!
  - 15th of October: quiz from the reading material
DIGITAL COMPUTATION
A computer

Devices

Gates

Transistors
Everything in a PC is Binary
... well, almost ...

<table>
<thead>
<tr>
<th>States of a Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
A transistor

This circuit functions as a switch. In other words, based on the control voltage, the circuit either passes Vin to output or not.
Examples of transistors

Replica of the first transistor

A set of transistors, depicting the fast change in technology.
A computer

Devices

Gates

Transistors
NOT Gate

<table>
<thead>
<tr>
<th>X</th>
<th>(\overline{X})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
### AND gate

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X·Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
OR Gate

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X+Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
# XOR Gate

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X ⊕ Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

![XOR Gate Diagram](image-url)