ELE 108 Overview of Computers & Programming

Ali Ziya Alkar
Mehmet Demirer
• **Text Book:**
  
• You will have irregular lab hours within the semester at the Computer Lab. in the department (see below).

• **Instructors:**
  Dr. Ali Ziya Alkar (section 21) Dr. Mehmet Demirer (section 22)

• **Phone:**
  Dr Ali Ziya Alkar  297 7027  Dr Mehmet Demirer  297 7025

• **e-mail:**
  alkar@hacettepe.edu.tr  mehmet@hacettepe.edu.tr

• **Course's Homework**

**Softcopy:**
• **Check the web page for the latest homework through the web page below. Submit your homework through internet.**
  
http://www.ee.hacettepe.edu.tr/~alkar/ELE108/labs

• **Hardcopy:**

-------------------------------

• **Grading**: Midterm %40, Final %40, Homeworks/Attendance/In Class Participation %5, Lab %15

• **Attendance** is required in all course hours and labs.
• Course Outline
  • 1. Overview of Computers and Programming
     1.1 Computer Architecture
     1.2 Program Development

  2. Overview of C (1.*, 2.1-2.6, 3.2, 7.1-7.2)
     2.1 C Language Elements
     2.2 Variable Declarations and Data Types
     2.3 Executable Statements
     2.4 General Form of a C Program
     2.5 Arithmetic Expressions
     2.6 Formatting Numbers in Program Output
     2.7 Library Functions
     2.8 Representation and Conversion of Numeric Types
     2.9 Representation and Conversion of Type char

  3. Selection Structures: IF and SWITCH Statements (4.*)
     3.1 Relational and Logical Operators
     3.2 if Statement
     3.3 Compound Statements
     3.4 Nested if Statements
     3.5 switch Statement
4. Repetition and Loop Statements (5.*)
4.1 while Statements
4.2 for Statements
4.3 do-while Statements
4.4 Nested Loops

5. Modular Programming (3.4-3.5, 6.*)
5.1 Functions without Arguments
5.2 Functions with Input Arguments
5.3 Functions with Simple Output Parameters
5.4 Scope of Names
5.5 Formal Output Parameters as Actual Arguments
6. Arrays (8.*)
6.1 Declaring and Referencing Arrays
6.2 Array Subscripts
6.3 Using for Loops for Sequential Access
6.4 Using Array Elements as Function Arguments
6.5 Arrays Arguments
6.6 Multidimensional Arrays

7. Strings (9.1-9.4)
7.1 String Basics and strcpy, strcat, strcmp, strchr, strstr
7.2 String Comparison
Collaboration Policy

• exams: no access to any material nor discussion with anyone (except the instructor) is allowed.

• assignments: solutions should be developed independently. Giving or receiving any code drawings, diagrams, text, or designs from another person is not allowed. Having access to another student’s work electronically or giving access is not allowed.

• max penalty for academic dishonesty: F in the course; reported to the university.
Outline

• A little History and intro
• Overview of Computers
  ▪ Hardware
  ▪ Memory
  ▪ Software
• Computer Languages
• Software Development Method
• Pseudo Code and Flowcharts
Computer History

• 3000BC-500BC
  Abacus
• 1642
  **Blaise Pascal**, a French religious philosopher and mathematician, builds the first practical mechanical calculating machine.
Computer History (Cont.)

• 1673
  Leibnitz invented Multiplication Machine
• 1830
  The "Analytical Engine" is designed by Charles Babbage
Computer History (Cont.)

• 1890

The U.S. Census Bureau adopts the Hollerith Punch Card, Tabulating Machine and Sorter to compile results of the 1890 census, reducing an almost 10-year process to 2 ½ years, saving the government a whopping $5 million. Inventor Herman Hollerith, a Census Bureau statistician, forms the Tabulating Machine Company in 1896. The TMC eventually evolved into IBM.
• **1939**
The first semi-electronic digital computing device is constructed by John Atanassoff.

• **1941**
German inventor Konrad Zuse produces the Z3 for use in aircraft and missile design but the German government misses the boat and does not support him.

• **1943**
English mathematician Alan Turing begins operation of his secret computer for the British military. It was used by cryptographers to break secret German military codes. It was the first vacuum tube computer but its existence was not made public until decades later.

• **Thomas Harold Flowers** built the first digital and programmable computer called the **Colossus**
Computer History (Cont.)

- First generation electronic computers
  - 1946
    - ENIAC (Electronic Numerical Integrator And Calculator)
    - 30 tons, 8 ft high, 30 ft long
    - Used thousands tubes & valves
  - 1951
    Univac I (Universal Automatic Computer), using a Teletype keyboard and printer for user interaction, and became the first commercially available computer. It could handle both numerical and alphabetic data.
Computer History (Cont.)

• 2nd Generation Computers (1954-59)
  ▪ Transistor invented by William Shockley at Bell Labs
  ▪ National Bureau of Standards (NBS) introduced its Standards Eastern Automatic Computer (SEAC)
  ▪ The first magnetic disk drive designed by Jacob Rabinow
  ▪ IBM introduced the 702 business computer in 1955
  ▪ Bendix G-15A small business computer sold for only $45,000, designed by Harry Huskey of NBS in 1956
Computer History (Cont.)

• 3rd Generation Computers (1959-71)
  ▪ Jack Kilby of Texas Instruments patented the first integrated circuit (IC) in Feb. 1959
  ▪ IBM announced the System/360 all-purpose computer, using 8-bit character word length (a "byte") in 1964
  ▪ DEC introduced the first "mini-computer", the PDP-8, in 1968
  ▪ Development began on ARPAnet, funded by the DOD in 1969
Computer History (Cont.)

• 4th Generation Computers (1971--?)
  ▪ Large Scale Integration (LSI) and VLSI
  ▪ Intel inc introduced the 4-bit 4004, a VLSI of 2300 components in 1971
  ▪ IBM developed the first true sealed hard disk drive, called the "Winchester" in 1973
Computer History (Cont.)

• 4th Generation Computers (1971--?)
  ▪ In 1980, IBM signed a contract with the Microsoft Co. of Bill Gates and Paul Allen and Steve Ballmer to supply an operating system for IBM's new PC model. Microsoft paid $25,000 to Seattle Computer for the rights to QDOS that became Microsoft DOS, and Microsoft began its climb to become the dominant computer company in the world.
  ▪ Apple Computer introduced the Macintosh personal computer 1984
Computer History (Cont.)

- Fifth Generation Computer (?)
  bio-computer?
Category

- **Personal computer**
  - Used by a single person at a time.
- **Mainframes**
  - Large real-time transaction processing systems
- **Supercomputer**
  - Largest capacity and fastest mainframes
Computer Components

- Hardware
equipment to perform computations
- Software
programs used to executed on a computer by providing the instructions.
Computer Hardware

• Main Memory
  ROM, RAM, etc.

• Secondary Memory
  Hard disk, floppy disk, CD, DVD, zip etc.

• Central Processing Unit (CPU)

• Input devices
  keyboard, mouse, scanners etc.

• Output Devices
  monitors, printers etc.
Components of a Computer

Secondary storage

Input devices

Main memory

Central processing unit

Output devices
Main Memory

- Memory cell
- Address
- Bytes and bits
- Store and retrieve

<table>
<thead>
<tr>
<th>Address</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-27.2</td>
</tr>
<tr>
<td>1</td>
<td>354</td>
</tr>
<tr>
<td>2</td>
<td>0.005</td>
</tr>
<tr>
<td>3</td>
<td>-26</td>
</tr>
<tr>
<td>4</td>
<td>H</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>998</td>
<td>X</td>
</tr>
<tr>
<td>999</td>
<td>75.62</td>
</tr>
</tbody>
</table>
Hardware & Software

• Hardware is the equipment used to perform the necessary computations.
  ▪ i.e. CPU, monitor, keyboard, mouse, printer, speakers etc.

• Software consists of the programs that enable us to solve problems with a computer by providing it with a list of instructions to follow
  ▪ i.e. Word, Internet Explorer, Linux, Windows etc.
Computer Hardware

• Main Memory
  ▪ RAM - Random Access Memory - Memory that can be accessed in any order (as opposed to sequential access memory), volatile.
  ▪ ROM - Read Only Memory - Memory that cannot be written to, no-volatile.

• Secondary Memory - Hard disks, floppy disks, zip disks, CDs and DVDs.

• Central Processing Unit - Coordinates all computer operations and perform arithmetic and logical operations on data.

• Input/Output Devices - Monitor, printer, keyboard, & mouse.

• Computer Networks – Computers that are linked together can communicate with each other. WAN, LAN, MAN, Wireless-LAN.
Memory

- Memory Cell (MC) – An individual storage location in memory.
- Address of a MC- the relative position of a memory cell in the main memory.
- Content of a MC – Information stored in the memory cell. e.g Program instructions or data.
  - Every memory cell has content, whether we know it or not.
- Bit – The names comes from binary digit. It is either a 0 or 1.
- Byte - A memory cell is actually a grouping of smaller units called bytes. A byte is made up of 8 bits.
  - This is about the amount of storage required to store a single character, such as the letter H.
1000 memory cells in Main memory

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-27.2</td>
</tr>
<tr>
<td>1</td>
<td>354</td>
</tr>
<tr>
<td>2</td>
<td>0.005</td>
</tr>
<tr>
<td>3</td>
<td>-26</td>
</tr>
<tr>
<td>4</td>
<td>H</td>
</tr>
<tr>
<td>998</td>
<td>X</td>
</tr>
<tr>
<td>999</td>
<td>75.62</td>
</tr>
</tbody>
</table>
Relationship Between a Byte and a Bit
Main Memory (Cont.)

- Random access memory (RAM)
  temporary storage of programs and data, the contents will be eliminated when the computer is off
- Read-only memory (ROM)
  store program or data permanently, used to store startup and critical instructions.
Secondary Storage

- Main memory is small and expensive
- Secondary memory: large and inexpensive
  - floppy disks
  - tapes
  - hard disks
  - CDs
  - DVDs
  - zip
  - jump drive
CPU

• Coordinating all computer operations and performing arithmetic and logical operations on data
• Fetch instructions and perform the actual manipulation
• Modern CPU is housed in a single integrated chip. (it also hosts registers, caches, etc.)
Computer Network

- Local Area Network (LAN)
- Metropolitan Area Network (MAN)
- Wide Area Network (WAN)

Internet access methods:
  - Dial up, DSL, Cable Modem and network cable
Internet Evolution

• ARPANET goes online in 1969.
• Bob Kahn and Vint Cerf develop the basic ideas of the Internet in 1973.
• In 1974 BBN opens the first public packet-switched network - Telenet.
• A UUCP link between the University of North Carolina at Chapel Hill and Duke University establishes USENET in 1979.
• TCP/IP (Transmission Control Protocol and Internet Protocol) is established as the standard for ARPANET in 1982.
Internet Evolution (Cont.)

• 1987: the number of network hosts breaks 10,000.
• 1989: the number of hosts breaks 100,000.
• 1992: the number of hosts breaks 1,000,000.
• The World Wide Web sports a growth rate of 341,634% in service traffic in its third year, 1993.
• The number of hosts today are nearly 275,000,000.
A Wide Area Network with Satellite Relays of Microwave Signals
Computers

- Computers receive, store, process, and output information.
- Computer can deal with numbers, text, images, graphics, and sound.
- Computers are worthless without programming.
- Programming Languages allow us to write programs that tells the computer what to do and thus provides a way to communicate with computers.
- Programs are then converted to machine language (0 and 1) so the computer can understand it.
Computer Software

• Operating System - controls the interaction between machine and user. Example: Windows, Unix, Dos etc.
  ▪ Communicate with computer user.
  ▪ Manage memory.
  ▪ Collect input/Display output.
  ▪ Read/Write data.

• Application Software - developed to assist a computer use in accomplishing specific tasks. Example: Word, Excel, Internet Explorer.
Computer Languages

• Machine Language – A collection of binary numbers
  ▪ Not standardized. There is a different machine language for every processor family.

• Assembly Language - mnemonic codes that corresponds to machine language instructions.
  ▪ Low level: Very close to the actual machine language.

• High-level Languages - Combine algebraic expressions and symbols from English
  ▪ High Level : Very far away from the actual machine language
  ▪ For example: Fortran, Cobol, C, Prolog, Pascal, C#, Perl, Java.
## Example

<table>
<thead>
<tr>
<th>Memory addresses</th>
<th>Machine Language Instructions</th>
<th>Assembly Language Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000000</td>
<td>00000000</td>
<td>CLA</td>
</tr>
<tr>
<td>000000001</td>
<td>00010101</td>
<td>ADD A</td>
</tr>
<tr>
<td>000000010</td>
<td>00010110</td>
<td>ADD B</td>
</tr>
<tr>
<td>000000011</td>
<td>00110101</td>
<td>STA A</td>
</tr>
<tr>
<td>000000100</td>
<td>01110111</td>
<td>HLT</td>
</tr>
<tr>
<td>000000101</td>
<td>?</td>
<td>A ?</td>
</tr>
<tr>
<td>000000110</td>
<td>?</td>
<td>B ?</td>
</tr>
</tbody>
</table>
High-level language

- Easy to write, easy to understand
- Can not be executed directly
- Compiler is used to convert high-level language into the target computer’s machine language
- Linker is used to generate the executable program
Some concepts

- compiler
- Source file
- Syntax
- Object file
- Linker
- Integrated development environment (IDE)
- Input data
- Program output
Word Processor (editor) Used to type in program and corrections

Source File
Format: text

Compiler
Attempts to translate program into machine code

Successful

Object File
Format: binary

Other Object Files
Format: binary

Linker
Resolves cross-references among object files

Executable File
Load module
Format: binary

Loader
Copies executable file into memory; initiates execution of instructions

Input data

Results
Example of Computer Languages

C Source Code:

```c
char name[40];
printf("Please enter your name\n");
scanf("%s", name);
printf("Hello %s", name);
```

Assembly Code:

```assembly
push    offset string "Please enter your name\n" (41364Ch)
call    dword ptr __imp__printf (415194h)
add     esp,4
lea     eax,[name]
push    eax
push    offset string "%s" (413648h)
call    dword ptr __imp__scanf (41519Ch)
add     esp,8
lea     eax,[name]
push    eax
push    offset string "Hello %s" (41363Ch)
call    dword ptr __imp__printf (415194h)
add     esp,8
```

Machine Code:

```
68 4C 36 41 00 FF 15 94 51 41 00 83 C4 04 8D 45 D8
50 68 48 36 41 00 FF 15 9C 51 41 00 83 C4 08 8D 45
D8 50 68 3C 36 41 00 FF 15 94 51 41 00 83 C4 08
```
C & Java: Source to execution

• For C, the generation and execution appear as in the following diagram
  
  C source code → compiler (program) → object code → linking and loading (program) → Executable

• Java programs do not ordinarily go through this set of steps. Java programs are "executed" by an interpreter; in a diagram it appears as
  
  Java source code → compiler (program) → Java byte codes → interpreter (program)

• More specifically, the compilation of C programs appears in diagram as
  
  C source code → preprocessor (program) → compiler (program) → lining and loading (program) → program execution
Flow of Information During Program Execution

Input data: meter readings

Step 1
Program input

Memory

Machine language program for computing water bill

Data entered during execution

Central processing unit

Computed results

Program output

Output results: water bill

Step 2

Step 3
Software Development Method

1. Specify **problem** requirements
2. Analyze the problem
3. Design the algorithm to solve the problem
4. Implement the algorithm
5. Test and verify the completed program
6. Maintain and update the program
Steps Defined

1. **Problem** - Specifying the problem requirements forces you to understand the problem more clearly.

2. **Analysis** - Analyzing the problem involves identifying the problem’s inputs, outputs, and additional requirements.

3. **Design** - Designing the algorithm to solve the problem requires you to develop a list of steps called an **algorithm** that solves the problem and then to verify the steps.

4. **Implementation** - Implementing is writing the algorithm as a program.

5. **Testing** - Testing requires verifying that the program actually works as desired.

6. **Maintenance** - Maintaining involves finding previously undetected errors and keep it up-to-date.
Converting Miles to Kilometers

1. **Problem**: Your summer job wants you to convert a list of miles to kilometers. You’re too lazy to do this by hand, so you decide to write a program.

2. **Analysis**
   - We need to get miles as input
   - We need to output kilometers
   - We know 1 mile = 1.609 kilometers

3. **Design**
   1. Get distance in miles
   2. Convert to kilometers
   3. Display kilometers
4. Implementation

```c
/*
 * Converts distance in miles to kilometers.
 */

#include <stdio.h> /* printf, scanf definitions */
#define KMS_PER_MILE 1.609 /* conversion constant */

int
main(void)
{
    double miles, /* input - distance in miles. */
            kms; /* output - distance in kilometers */

    /* Get the distance in miles. */
    printf("Enter the distance in miles> ");
    scanf("%lf", &miles);

    /* Convert the distance to kilometers. */
    kms = KMS_PER_MILE * miles;

    /* Display the distance in kilometers. */
    printf("That equals %f kilometers.\n", kms);

    return (0);
}

Sample Run
Enter the distance in miles> 10.00
That equals 16.090000 kilometers.
```
5. Test

- We need to test the previous program to make sure it works. To test we run our program and enter different values and make sure the output is correct.

6. Maintenance

- Next summer, your boss gets a contract with NASA, so he wants you to add support for converting to AU’s
Pseudo code & Flowchart

• **Pseudo code** - A combination of English phrases and language constructs to describe algorithm steps

• **Flowchart** - A diagram that shows the step-by-step execution of a program.

• **Algorithm** - A list of steps for solving a problem.
Why use pseudo code?

• Pseudo code cannot be compiled nor executed, and there are no real formatting or syntax rules.
• It is simply one step - an important one - in producing the final code.
• The benefit of pseudo code is that it enables the programmer to concentrate on the algorithms without worrying about all the syntactic details of a particular programming language.
• In fact, you can write pseudo code without even knowing what programming language you will use for the final implementation.
• Example:
  
  Input Miles
  Kilometers = Miles * 1.609
  Output Kilometers
Another Example of Pseudo code

- **Problem**: Calculate your final grade for CS 201
- **Specify the problem** - Get different grades and then compute the final grade.
- **Analyze the problem** - We need to input grades for exams, labs, quizzes and the percentage each part counts for. Then we need to output the final grade.
- **Design**
  1. Get the grades: quizzes, exams, and labs.
  2. Grade = .30 * 2 regular exams & quizzes + .20 * Final exam + .50 * labs
  3. Output the Grade
- **Implement** – Try to put some imaginary number and calculate the final grade after you learn how to program.
Flowcharts

Flowchart uses boxes and arrows to show step by step execution of a program.

- **Process**
- **Decision**
- **Display**
- **Start or Terminal**
- **Document**
- **Manual Input**
Example of a Flowchart

Start

Get Grades and percentages

Calculate Final grade

Display Grade

End
Compiler

- Compilation is the process of translating the source code (high-level) into executable code (machine level).
- Source file - A file containing the program code
  - A Compiler turns the Source File into an Object File
- Object file - a file containing machine language instructions
  - A Linker turns the Object File into an Executable
- Integrated Development Environment (IDE) - a program that combines simple word processing with a compiler, linker, loader, and often other development tools
  - For example, Eclipse or Visual Studio
Fig 1.12
Entering, Translating, and Running a High-Level Language Program