Computer Programming Using C
COP 3275 - Summer 2017

Lecture 1: Introduction

By Ahmed E. Khaled
“Computer Science”?

• CS is a discipline that involves understanding and designing of computers and computational processes that deal with software systems to solve problems.

• With a central focus of handling / manipulating information, many problems in healthcare, business, and other areas can be solved effectively with computers and existence of domain knowledge.
“Computer”?

- Simply a set of *small electronic switches* that are either on or off!

- With different combinations of these switches, you can make the computer do something (e.g. display something).
Computers use a simple language (Known as *Machine Language*) that consists of only 1s and 0s, where 1 means "on" and 0 means "off."
• Trying to talk to a computer in Machine Language is hard!, thus a programming language acts as a *translator* between you and the computer that simply telling a computer what to do.

• Specialized program known as a *compiler* takes the instructions written in the programming language and converts them to machine language.
“Programming Language”? 

- PL is a special language used by programmers to *develop programs* (sets of instructions) that involve a computer to perform some kind of computation.

- A program does nothing more than *tell the computer* to accept input, manipulate it, and output results in some form.

- As any language, PL is split into two components:
  - *Syntax*: defines the form and rules to write program.
  - *Semantics*: defines the meaning and logic behind.
## Examples on programs!

<table>
<thead>
<tr>
<th>Program</th>
<th>Program Inputs</th>
<th>What the Program Does</th>
<th>Program Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Word Editor</strong></td>
<td>Characters you type from keyboard</td>
<td>Formats the text and corrects spelling</td>
<td>Displays / prints neatly organized text</td>
</tr>
<tr>
<td><strong>Game</strong></td>
<td>Keystrokes or joystick movements</td>
<td>Calculates how fast and far to move the figure on-screen</td>
<td>Moves a cartoon figure on-screen</td>
</tr>
<tr>
<td><strong>Stock Market Predictor</strong></td>
<td>Current and past prices for stocks</td>
<td>Tries to recognize trends in a stock's price fluctuations</td>
<td>Predicts the future price of a stock</td>
</tr>
</tbody>
</table>
Application programs
Word, games, database

Systems Software
Operating systems, device drivers, compilers

Machine with all hardware components
Logical Units of a Computer System:

- **Input unit** (e.g. Mouse, keyboard)
- **Output unit** (e.g. Printer, monitor, audio speakers)
- **Arithmetic and logic unit (ALU)**: Performs calculations
- **Memory unit**: Stores the *running program*, retains input and processed information
- **Central processing unit (CPU)**: *execute instructions* and Supervises operation of other devices
- **Secondary storage unit (e.g. Hard-drives)**: store data and programs that are *not* running at that time.
C

• C is a by-product of the UNIX operating system, which was developed at Bell Laboratories by Ken Thompson and others in the late 1960s.

• UNIX was written in assembly language, low-level programming language with a very strong correspondence between the language and the hardware architecture.

• Programs written in assembly language are usually painful to develop and debug (small set of instructions).
• Thompson decided that a higher level language was needed for the further development of UNIX.

• The most important feature of the high-level programming languages is the portability.

• Portable means the developed program can be adapted, so that an executable version can be created for totally different computing environments that are differ from the original one (where the program was originally designed).

• Portability is done by a compiler.
High-level languages (HLL)

• HLL are platform independent where you can write a program once and run it in different types of machines.

• A program written in a high-level language is called a source program, that must be translated into machine code for execution. The translation can be done using another programming tool called an interpreter or a compiler.
  • Interpreter reads one statement from the source program at a time, translates it to the machine code, and then executes it right away.
  • Compiler translates the entire source program into a machine-code, and the machine-code file is then executed.
C takes a middle path between:

- **low-level assembly** language
  - Direct access to memory layout through pointer manipulation

- **high-level programming** language like Java:
  - Block structure
  - Some encapsulation of code, via functions
• The standard for C language was completed in December 1989, while other language based on C started to evolve (e.g., C++).

• C++ added more features to support object-oriented and modular programming (we may come across from time to time).

• C++ facilitates large software development projects.
Strengths of C

• C provides access to *machine-level concepts* (e.g., memory addresses) that other programming languages try to hide.

• C provides a limited set of features to keep it as a *small language*, and relies heavily on “*libraries*” of functions.

• C is very powerful, was originally designed for systems programming and now is used for *applications of all kinds* (e.g., embedded systems, commercial data processing).

• C compilers are *small and easily written*, which has helped make them widely available (portable).

• **C’s Weaknesses arise from the same source as many of its strengths, the closeness to the machine.**
C development environment includes

*System libraries and headers*

*Application Source*

*Preprocessor and Compiler*: converts your application to object code for a specific platform

*Linker*: resolves external references, ties object files and produces the executable module
Summary

• Computer switches
• Machine language
• Low-level vs high-level language
• Compiler
• Portability