Introduction to C
Computer?

- A programmable electronic device that can store, retrieve, and process data (Merriam-Webster Dictionary)
- A “black box” that will accept input and produce output
  - The output that is produced depends on the current program in the block box
The von Neumann Model

- Such computers have input and output devices, a main memory (RAM), and a CPU (the central processing unit).
- Main memory contains both the program and the data with which the program is working (program as data).

[Diagram of a computer system with CPU, RAM, mouse, keyboard, and monitor connected]

Computer Principles
Program?

- A set of instructions telling the computer what to do with the input in order to produce the output
- Machine code
  - The representation of a computer program which is actually read and interpreted by the computer
  - A program in machine code consists of a sequence of machine instructions
  - Instructions are binary strings
Programmimg Language?

- A formal language in which computer programs are written
- The definition of a particular language consists of both syntax (how the various symbols of the language may be combined) and semantics (the meaning of the language constructs)
- C, C++, Java, FORTRAN, Lisp, Scheme, ML, etc.
- Assembly language (machine dependent)
File?

- A complete collection of data treated by a computer as a unit especially for purposes of input and output (Merriam-Webster Dictionary)

- The contents are bit strings
  - Interpretation depends on the format of the file

- Text file
  - Ordinary ASCII or EBCDIC representation

```
011101111
010101000
111000111
011010001
000100100
```

011101111
010101000
111000111
011010001
000100100
Operating Systems

- Software that controls the operation of a computer and directs the processing of programs (as by assigning storage space in memory and controlling input and output functions)
- Windows, Linux, Unix, DOS, …
A program that converts another program from some source language (or programming language) to machine language (object code)
Text Editor?

- A utility program for creating and modifying text files
  - GNU emacs
  - UNIX vi
ANSI/ISO C Standard

- American National Standards Institute (ANSI)
- International Organization for Standardization (ISO)
- ANSI/ISO C is an internationally recognized standard
  - Almost all C compilers follow the standard
  - To promote portability, reliability, maintainability, and efficient execution of C language program on a variety of machines
Typical Compilation Phases

1. **Preprocessing**
   - foo.c
   - preprocessed C source

2. **Compilation**
   - gcc
   - foo.o

3. **Linking**
   - libraries
   - foo
```
#include <stdio.h>

int main(void)
{
    printf("Programming Made Easy\n");
    return 0;
}
```
Preprocessing

- **C preprocessor (cpp)**
  The standard Unix macro-expansion utility run as the first phase of the C compiler
- `#include <stdio.h>` causes the preprocessor to include a copy of the header file “stdio.h” at that point
  - `stdio.h` contains the information for `printf`
Functions

- Similar to mathematical functions but much less strict
  - A function may return different values each time it is called with the same argument values and may have side effects
- Side effect – modification of the state of the system
  - Modifying the contents of memory locations, writing data to a display or file, reading some data from other side-effecting functions, etc.
Main() Function

- Every program has this function
- Program execution always starts with this function
  - Conceptually, the OS calls main() to execute the program
- int main(void)
  - Takes no argument
  - Returns a value of type int
The C system contains a standard library of functions that can be used in programs

- `printf()` is defined in the library

- The `stdio.h` (header file) contains information about this function

- Prints its argument on the screen
A Simplistic Program Translation Scheme

- Problems:
  - Efficiency: small change requires complete recompilation
  - Modularity: hard to share common functions (e.g., printf)
A Better Scheme Using a Linker

```
  p.c  
  |    |  
  ↓    ↓
compiler
  |    |  
  ↓    ↓
p.o  
  |    |  
  ↓    ↓
linker
  |    |  
  ↓    ↓
p
```

```
  a.c  
  |    |  
  ↓    ↓
compiler
  |    |  
  ↓    ↓
a.o  
  |    |  
  ↓    ↓
p
```
Running the Executable Program

- The OS loader loads the application program into the main memory and initialize execution environment
- Control is transferred to the function main()
  - Execution starts
- `return` in main returns control to the OS
```bash
fedoraduo:/home/jlee/vwd/pme/day01> gcc -o foo foo.c
fedoraduo:/home/jlee/vwd/pme/day01> ls
foo  foo.c
fedoraduo:/home/jlee/vwd/pme/day01> foo
Programming Made Easy
fedoraduo:/home/jlee/vwd/pme/day01> ```
Computing?

- By human
  - Needs a pencil and a piece of paper
  - Following a sequence of computing steps (procedural)
  - Recording the intermediate results on the paper (imperative)

- C is an imperative and procedural language
  - Specifies explicit manipulation of the state of the computer system (imperative)
  - Specifies an explicit sequence of steps to perform (procedural)
Variables and Assignments

- **Variable**
  - A named memory location in which a program can store intermediate results and from which it can read them.

- **Assignment**
  - Storing the value of an expression in a variable.

```
x  321
y  32
z  15

x = y + z
```

```
x  47
y  32
z  15
```
## Characters in C Programs

- A C program is constructed by the programmer as a sequence of characters.
- Characters are collected by the compiler into syntactic units (tokens).

<table>
<thead>
<tr>
<th>Characters</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowercase letters</td>
<td>a, b, c, …, z</td>
</tr>
<tr>
<td>Uppercase letters</td>
<td>A, B, C, …, Z</td>
</tr>
<tr>
<td>Digits</td>
<td>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</td>
</tr>
<tr>
<td>Other characters</td>
<td>+, -, *, /, =, (, ), {, }, [, ]</td>
</tr>
<tr>
<td></td>
<td>&lt;, &gt;, ‹, “, !, #, %, &amp;, _,</td>
</tr>
<tr>
<td></td>
<td>^, ~, , , , ;, :</td>
</tr>
<tr>
<td>White space characters</td>
<td>blank, newline, tab, etc.</td>
</tr>
</tbody>
</table>
The syntax of C will be described using Backus-Naur Form (used in ALGOL 60)

<table>
<thead>
<tr>
<th>italics</th>
<th>Syntactic categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>::=</td>
<td>To be rewritten as</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>{}</td>
<td>grouping</td>
</tr>
<tr>
<td>{}+</td>
<td>Repeat 1 or more times</td>
</tr>
<tr>
<td>{}*</td>
<td>Repeat 0 or more times</td>
</tr>
<tr>
<td>{}opt</td>
<td>optional</td>
</tr>
</tbody>
</table>
Syntax Rules (contd.)

- `letter_or_digit ::= letter | digit`
- `letter ::= lowercase_letter | uppercase_letter`
- `lowercase_letter ::= a | b | ... | z`
- `uppercase_letter ::= A | B | ... | Z`
- `digit ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9`
- `alphanumeric_string ::= letter_or_digit *`
- `conditional_statement ::= if ( expression ) statement { else statement }_{opt}`
Comments

- Comments are arbitrary strings of symbols placed between the delimiters /* and */
  - Not tokens
- Compiler changes each comment into a single blank character

/* this is a comment */

/*************************************************************************

** This is a comment **

*************************************************************************/
Keywords

- Have a strict meaning as individual tokens
- Reserved (cannot be redefined or used in other contexts)

<table>
<thead>
<tr>
<th>auto</th>
<th>break</th>
<th>case</th>
<th>char</th>
<th>const</th>
<th>continue</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>do</td>
<td>double</td>
<td>else</td>
<td>enum</td>
<td>extern</td>
</tr>
<tr>
<td>float</td>
<td>for</td>
<td>goto</td>
<td>if</td>
<td>int</td>
<td>long</td>
</tr>
<tr>
<td>register</td>
<td>return</td>
<td>short</td>
<td>signed</td>
<td>sizeof</td>
<td>static</td>
</tr>
<tr>
<td>struct</td>
<td>switch</td>
<td>typedef</td>
<td>union</td>
<td>unsigned</td>
<td>void</td>
</tr>
<tr>
<td>volatile</td>
<td>while</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Identifiers

- **Syntax**
  - `identifier ::= { letter | _ } {letter | _ | digit }*`

- Identifiers are created to give unique names to objects in a program
  - Keywords – reserved identifiers
  - Variable names, function names

- **Case sensitive**

- Identifiers that begin with an underscore can conflict with system names

- In ANSI C, the first 31 characters are discriminated
Identifiers (contd.)

- i
- _id_ l
- computer_principles
- __computer_principles
Character Constants

- Enclosed in a pair of single-quote marks
  - ‘a’
  - ‘b’
  - ‘\n’

- Escaping the usual meaning of \n (\: the escape character)
- A single character called newline
String Constants

- A sequence of characters enclosed in a pair of double-quote marks
  - “abc”
  - “abc”\”def”
  - “abc\\def”

- Two string constants that are separated by white space are concatenated by the compiler (cpp) into a single string
  - “abc” “def” is equivalent to “abcdef”