Lecture 1.1

1. Computer and Computer Software – Brief
2. Computer Program Creation & Program Development Process – Brief
3. History of C (Computer Programming) Language
4. Components of a C Program – Simple Example

____

1. Computer and Computer Software – Brief

Computers have offered a great deal of time saving computations and applications from the 1950s to the current time. Computers have changed from a 30 tons and occupying 1500 square feet space to a very small size desktop box, laptop, tablet, or mobile/handheld devices.

The speed of today's computers is in no comparison to any of the past and it is getting to be more powerful every day.

- Different computer architectures and chipsets are being used or developed for the next generation of computers.
- All peripheral components are getting better and cheaper so that computers are making their ways to the general public more and more every day.
- In many cases, computers become the necessities and nothing would go on or even possible without the assistance from computers.

Read with your own measure and pleasure – http://en.wikipedia.org/wiki/Computer

Computer –

However, without software to control/operate a given computer, it would just sit and become useless. So, when referring to our computer(s), we usually meant a system with an operating system and software installed to provide the operations and required functionalities for that computer.

Computer software must be developed and maintained. A piece of software may be very simple such as a utility or very sophisticated such as a complete operating system. No matter what the software was intended for, it needs to be developed and implemented.

And as time goes on, the software may need to be updated or upgraded to fix bugs or to take advantage of new components (existing hardware and/or software).

Definitely, computer programs and computer programming are the large parts of the solutions for many modern systems and applications.

Thus as demanding and demanded,

- To create a computer program, one would need to have tools and to learn proper techniques or ways when using these tools.
- A computer program can be written in a specific computer programming language to take advantage of each language in handling data and designing the solution.

Read with your own measure and pleasure – http://en.wikipedia.org/wiki/Computer_software

Where and how would the development of a piece of computer program or software start?

Let’s briefly look at the tools and general process of computer programming development in the next section.

2. Computer Program Creation & Program Development Process – Brief

What is computer programming?

In general, it is said that computer programming is the technique that uses computer to manipulate some given data values in order to produce the desired results.

One would write a program or code to manipulate, which are provided by the users, to produce the required output in the most efficient way.

The manipulation process will follow some specific sets of instructions or steps, which are then language-specific while being implemented.

So then, a computer program will provide instructions for a computer to manipulate the data and produce the output.

In general, the process of developing a computer program would require one to

1. Get/specify problem statement
2. Analyze problem
3. Design/develop algorithms
4. Implement algorithms
5. Test and debug program
6. Maintain and update program

Figures 1 and 2 below depict the process; and so then let's explain and elaborate on these steps.

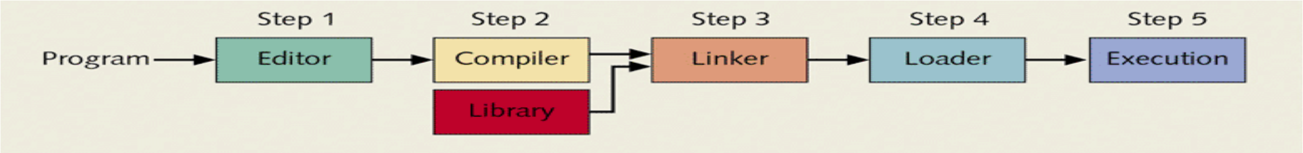

Figure 1 Creating and processing a high-level language program

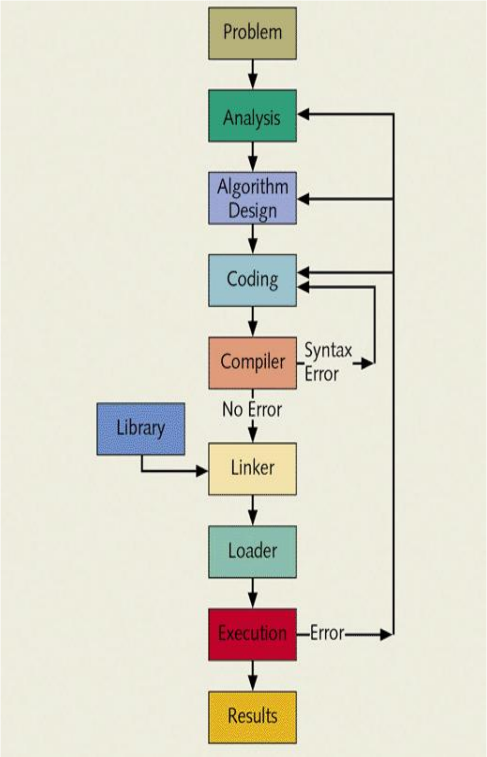

Figure 2 Development process – Problem analysis-design-coding-execution cycle

2.1 Get/Specify Problem Statement

This requires the problem to be stated clearly and formed in an unambiguous manner. Doing that would provide a definitive understanding of what is required for the solution and also to eliminate confusion.

2.2 Analyze Problem – The WHATs?

Analyzing the problem statement would break things into:

a) What the input would be, and

b) What the output would be, and

c) What are, if any, additional requirements or constraints to achieving the solution?

2.3 Design/Develop Algorithms

One may need to follow a sequence or list of steps to produce the solution. This list is combined into a so-called **algorithm**.

That means **HOW-TO** do things or **HOW-TO** process things. One must follow through with each of the steps and verify that the algorithm would indeed provide the desired outcome.

2.4 Implementation

Using a specific computer language, one can just translate the steps into program statements. Group them in methods/functions or objects and use these methods/functions and objects as necessary.

2.5 Test & Debug Program

A major portion of effort in developing computer code is spent in testing and debugging. The goal is to have the program working. There may be errors that need to be corrected; these can either be syntax errors or logic errors. These errors must be fixed by looping back to Step 4.

Generally, the logical errors are called bugs. Debugging really means trying to get rid of these logical errors.

2.6 Maintain & Update

After getting rid of all bugs and verifying the results, the program will be either installed or used. From time to time, this program may need to be relocated or reinstalled; it may also need to be rewritten (updated) to accommodate new conditions or systems.

2.7 Examples

Example 1

```c
/**
 * Program Name: cis6L0111.c
 * Discussion:   C Elements
 * Written By:   T.N.
 */
#include <stdio.h>

int main() {
  printf("Welcome to C World!\n");

  return 0;
}
```

OUTPUT

```
Welcome to C World!
```

Example 2

```
/**
 * Program Name: cis6L0112.cpp
 * Discussion:   C++ Elements
 * Written By:   T.N.
 */
#include <iostream>
using namespace std;

int main() {
  cout << "Welcome to C++ World!” << endl;

  return 0;
}
```

OUTPUT

```
Welcome to C++ World!
```

Example 3

```
/**
 * Program Name: HelloWorldApp.java
 * Written By:   T. N.
 * Discussion:   Java Elements
 */

class HelloWorldApp {
  public static void main(String[] args) { // Required
    System.out.println("Hello World!");    // Using System packages

    return;
  }
}
```

OUTPUT

```
Hello World!
```

What are so different with the above programs?

3. History of C (Computer Programming) Language – Brief

Excerpt From Wikipedia.org – Search: C (Programming Language Dated 2019/02/20)

> “**C** (/siː/, as in the letter *c*) is a general-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion, while a static type system prevents many unintended operations. By design, C provides constructs that map efficiently to typical machine instructions, and therefore it has found lasting use in applications that had formerly been coded in assembly language, including operating systems, as well as various application software for computers ranging from supercomputers to embedded systems.
>
> C was originally developed by Dennis Ritchie between 1972 and 1973 at Bell Labs. It was created to make utilities running on Unix. Later, it was applied to re-implementing the kernel of the Unix operating system[6]. Through 1980s, C gradually gained popularity. Nowadays, it is one of the most widely used programming languages[7][8] with C compilers from various vendors available for the majority of existing computer architectures and operating systems. C has been standardized by the American National Standards Institute (ANSI) since 1989 (see ANSI C) and subsequently by the International Organization for Standardization (ISO).
>
> …
>
> **Relations to other languages**

> Many later languages have borrowed directly or indirectly from C, including C++, C#, Unix's C shell, D, Go, Java, JavaScript, Limbo, LPC, Objective-C, Perl, PHP, Python, Rust, Swift, Verilog and SystemVerilog (hardware description languages).[5] These languages have drawn many of their control structures and other basic features from C. Most of them (with Python being the most dramatic exception) are also very syntactically similar to C in general, and they tend to combine the recognizable expression and statement syntax of C with underlying type systems, data models, and semantics that can be radically different. …"

In 1972 while collaborated on designing the Unix operating system, Dennis Ritchie of Bell Labs created C as a computer programming language. By the late 1970s, C had evolved and become more useful and powerful; it was known then as "traditional C".

In 1978, Kernighan and Ritchie wrote the first edition of the book titled "**The C Programming Language**" and drew wide attention to the language. This book was then updated with the second edition some time later and it is still being used today as required text and reference in many universities and industries.

- In 1989, the ANSI C (C89) standard was approved and the ISO (C90) was also approved later in 1990. These two standards are essentially the same.
- Excerpt from Wikipedia.org – Search: C (Progrmaming Language); (Dated on 2017/02/11)

"…

C99

C99 (previously known as **C9X**) is an informal name for ***ISO/IEC 9899:1999***, a past version of the C programming language standard.[1] It extends the previous version (C90) with new features for the language and the standard library, and helps implementations make better use of available computer hardware, such as the IEEE 754-1985 arithmetic, and compiler technology.[2]

Design

C99 is, for the most part, backward compatible with C89, but it is stricter in some ways.[3]

In particular, a declaration that lacks a type specifier no longer has `int` implicitly assumed. The C standards committee decided that it was of more value for compilers to diagnose inadvertent omission of the type specifier than to silently process legacy code that relied on implicit **`int`**. In practice, compilers are likely to display a warning, then assume **`int`** and continue translating the program.

C99 introduced several new features, many of which had already been implemented as extensions in several compilers:[4]

- inline functions
- intermingled declarations and code: variable declaration is no longer restricted to file scope or the start of a compound statement (block), facilitating static single assignment form
- several new data types, including `long long int`, optional extended integer types, an explicit boolean data type, and a `complex` type to represent complex numbers

- variable-length arrays (although subsequently relegated in C11 to a conditional feature which implementations are not required to support)
- flexible array members
- support for one-line comments beginning with `//`, as in BCPL, C++ and Java
- new library functions, such as `snprintf`
- new headers, such as `<stdbool.h>`, `<complex.h>`, `<tgmath.h>`, and `<inttypes.h>`
- type-generic math (macro) functions, in `<tgmath.h>`, which select a math library function based upon `float`, `double`, or `long double` arguments, etc.
- improved support for IEEE floating point
- designated initializers
- compound literals (for instance, it is possible to construct structures in function calls: `function((struct x) {1, 2})`)[5]
- support for variadic macros (macros with a variable number of arguments)
- `restrict` qualification allows more aggressive code optimization, removing compile-time array access advantages previously held by FORTRAN over ANSI C[6]
- universal character names, which allows user variables to contain other characters than the standard character set
- keyword `static` in array indices in parameter declarations[7]

Parts of the C99 standard are included in the current version of the C++ standard, C++11, including integer types, headers, and library functions. Variable-length arrays are not among these included parts because C++'s Standard Template Library already includes similar functionality.

C11

C11 (formerly **C1X**) is an informal name for ***ISO/IEC 9899:2011***,[1] the current standard for the C programming language. It replaces the previous C standard, informally known as C99. This new version mainly standardizes features that have already been supported by common contemporary compilers, and includes a detailed memory model to better support multiple threads of execution. Due to delayed availability of conforming C99 implementations, C11 makes certain features optional, to make it easier to comply with the core language standard.[2][3]

C18

Main article: C18 (C standard revision)

Published in June 2018, C18 is the current standard for the C programming language. It introduces no new language features, only technical corrections and clarifications to defects in C11. The standard macro `__STDC_VERSION__` is defined as `201710L`.

…”

- Many current compilers may not fully implement all features (which may not be of our concerns in this class).

It is important to repeat again here that

- ✦ The goal of computer programming is to work with some (data) model so that a design and then its implementation will produce a computer program that will operate on the data model.
- ✦ A computer program may work with one or more platforms (operating systems, embedded systems, etc) to produce the desired output and results.

Why C?

- C is a structural/procedural language that is powerful and flexible.
- C is also efficient and portable.
- OK To check out —

  http://www.cio.com/article/3169540/application-development/what-is-the-point-of-learning-c.html

  http://computer.howstuffworks.com/benefits-learning-c-programming.htm/printable

  http://www.cprogramming.com/whyc.html

  https://www.pluralsight.com/blog/software-development/why-every-programmer-should-learn-c

  http://www.streetdirectory.com/travel_guide/114363/programming/10_reasons___why_c_should_be_your_first_programming_language.html

  https://www.reddit.com/r/learnprogramming/comments/30pp5a/is_spending_the_time_to_fully_learn_c_worth_it/#bottom-comments

  https://forum.freecodecamp.com/t/learning-c-language-before-javascript/11407/3

  https://courses.edx.org/courses/course-v1:HarvardX+CS50+X/info

- Old But Interesting

  http://iel.ucdavis.edu/publication/WhyC.html

  (10 Reasons to Teach and Learn Computer Programming in C – 2011/05/30)

  https://www.softintegration.com/academic/why-learn-c-as-first-programming-language.html

- Fun Tools (At Least) --

  http://www.learn-c.org/

- One of the “Real” Tools (At Least) --

  Visual Studio

- Fun Stuff (To Look At) –

https://projecteuler.net/

https://code.google.com/codejam/

https://developers.google.com/

https://www.joelonsoftware.com/2005/01/02/advice-for-computer-science-college-students/

http://cs50.tv/2017/fall/

Khan Academy — https://www.khanacademy.org/computing/computer-science/how-computers-work2/v/khan-academy-and-codeorg-binary-data

Code.org — https://code.org/educate/resources/videos

https://mikkegoes.com/computer-science-binary-code-explained/

https://study.com/academy/lesson/machine-code-and-high-level-languages-using-interpreters-and-compilers.html

How do computers …? — https://www.youtube.com/watch?v=QXjU9qTsYCc

https://www.youtube.com/watch?v=HbgzrKJvDRw

https://www.youtube.com/watch?v=oxuRxtrO2Ag

UC Berkeley — http://cs10.org/sp19/discussion/02/Disc2.pdf

Why C Again – In Brief!!!

- C is natural beginning footstep to C++ (What is C++?) – Search and get impressed for yourself.
- "C is a concise, yet powerful, language well suited for microcontroller work." As excerpted from "Beginning C for Arduino" by Jack Purdum, Apress 2013.
- VEX Studio (also RobotC) is designed and targeting VEX Cortex ROBOT families or generations.
- C has been one of the many important tools used in embedded systems and applications development (graphic libraries, Linux, *nix, smart device apps, etc.).
- C will produce native executable code that would be extremely fast – Embedded systems and applications.
- Many many many syntax of C would be used in many many other modern programming languages – Search and get impressed for yourself (above links).

For our purpose in this class, of course, the computer programs and exercises will be written in C language (or just C).

4. Components of a C Program – Simple Example

Let us look at a C program below.

Example

```
/**
```

```
 * Program Name: cis6L0114.c
 * Discussion:   C Elements
 * Written By:   T.N.
 */
#include <stdio.h>

int main() {
  printf("Welcome to C World Again!\n");

  return 0;
}
```

OUTPUT

```
Welcome to C World Again!
```

The above C program has

- Two **statements**,
- Several built-in **keywords**, and
- A predefined **function** (or method). The function `printf()` was defined in `stdio.h`, which is the header file to be loaded and preprocessed by the **`#include`** directive.

What are the components that one can use to form a C **statement** and eventually produce a correct program?

A C statement may be formed by using a combination of

- Keywords (C vocabularies),
- User's defined identifiers such as variables and functions/methods,
- Constants, and
- Operators

4.1 Keywords

Keywords (vocabularies) are built-in terms (or words) that reserved for the language. They are defined to be used for specific purposes; thus they must be used as they were defined.

For examples, the `include` keyword is used to signal the compiler to preprocess the specified file (such as, `stdio.h`) prior to taking in the next segment of the code. The `int` keyword modifies the returned value of the method `main()` and forces it to be of type `int`. The `return` keyword is to indicate a returning process, which may also depend on external setup.

4.2 Identifiers

Identifiers are the names created in the code by the programmer. An identifier can be used as variable name, function name, etc. To be a valid identifier, it must start with a character and may have any combination of alphanumeric characters and some special characters, such as _ or `$`.

It is recommended to only use alphanumeric characters for identifiers. In addition, an identifier should have some relevant and meaningful interpretation for the tasks that are being written for.

Note that C is a **case-sensitive** language. That means the two identifiers of "**`yours`**" and "**`Yours`**" are two different specifications or references in the code.

4.3 Constants

Briefly, constants are values that cannot be changed during the execution of the program.

4.4 Executable Statements

An executable statement is a statement that will get compiled and/or may get linked with other modules or functions/methods.

In C, an executable statement is terminated with a semi-colon (;).

For examples,

```
printf("Welcome to C World Again!\n");
```

or,

```
return 0;
```

4.5 Comments

C program may also have non-executable statements (that means the compiler will ignore them during the compilation). They are called comments and used to provide explanations and reminders along the code.

A block comment will start and close with a pair of “/*” and “*/”. This style of commenting can span (have) several lines as long as these lines are between the given comment pair. Note that this comment style cannot be nested.

For examples,

Valid Comment

```
/**
 * Program Name: cis6L0114a.c
 * Discussion:   C Elements
 * Written By:   T.N.
 */
```

Invalid Comment

```
/**
 * Program Name: cis6L0114b.c
 * Discussion:   C Elements
 * Written By:   T.N.

 /*Laney College*/                    ← ERROR

 */
```

Some compilers (C99 and later on implementation) also provide a second style of commenting. This style uses a double forward slash pair of // to comment out the remaining text of the current line.

For examples,

```
// Laney College
```

Or,

```
printf(“Welcome to C World again!\n”); // using printf()
```

4.6 Braces and Blocks

We use the pairs of curly braces (that means ‘ { ’ and ‘ } ’ symbols) to specify (or delimit) functions and blocks/structures. We will have more discussion on this later.

Obviously, the above simple program produces the result, which is the output being displayed on screen. More specifically, the data (which is the phrase or string “`Welcome to C World again!\n`”) is being processed by the function (`printf()`, which is predefined – where?).

We may be asking questions such as:

- What can a function (or program) operate or work with? Or
- What kinds of data (data types) can we have and use in a computer program?

You should have access to at least one good C reference (book/reading resource) for this class!

Data are or data is?

Is it singular or plural? It's a word we use every day here on the Datablog - but are we getting it completely wrong?
https://www.theguardian.com/news/datablog/2010/jul/16/data-plural-singular
(2019/06/24)

Data

Data is often treated as a plural noun in writing related to science, mathematics, finance, and computing. Elsewhere, most English speakers treat it as a singular mass noun. This convention is well established and widely followed in both edited and unedited writing. Keep in mind, though, that some people consider the singular *data* incorrect. This view is based on a misunderstanding of how English develops, but those who hold it tend to feel strongly about it, so we might approach *data* with caution in writing for school or work.
https://grammarist.com/usage/data/
(2019/06/24)

Which is correct, data is or data are?

Parag Pandya, Programmer Analyst at Cognizant (2015-present)
Answered Mar 5, 2017

Definition of OXFORD dictionary ::

- In Latin, **data** is the plural of **datum.**
- Historically and in specialized scientific fields, it is treated as a plural in English e.g. data were collected and classified.

- In modern non-scientific use, it is often not treated as a plural. Instead, it is treated as a mass noun, similar to a word like information, which cannot normally have a plural and which takes a singular verb. Sentences such as data was (as well as data were) collected over a number of years are now widely accepted in standard English.

Office for National Statistics ::

The word data is a plural noun so write "data are". Datum is the singular.
https://www.quora.com/Which-is-correct-data-is-or-data-are
(2019/06/24)

Is the Word "Data" Singular or Plural?

Naomi Robbins

I help people communicate data clearly with graphs.

I recently attended an event where many of the other guests worked in corporate communications. A professor lamented that many of today's students brought up with text messages and Twitter TWTR +0% don't write well. Another attendee agreed, noting that on three recent occasions she has heard people say, "Data are." I chimed in, "What's wrong with that? The word 'data' is plural." That's what my supervisor in my early days in industry at Bell Laboratories insisted. That's what the copy editor of *Creating More Effective Graphs* said.
Here is a link that discusses this issue:
http://grammar.quickanddirtytips.com/is-data-singular-or-plural.aspx.
Charles Carson in a guest post for Grammar Girl describes *count nouns* and *mass nouns*. Count nouns answer *how many* while mass nouns answer *how much*. The simplified answer here is that data can be either, depending if it is a *count noun* (can be replaced by *facts*) or a *mass noun* (can be replaced by *information).* An example of data used as a count noun is, "The data consist of the names, heights, and weights of the 30 children in this class." An example of data used as a mass noun is, "Data is increasing at an incredible rate."
https://www.forbes.com/sites/naomirobbins/2012/07/25/is-the-word-data-singular-or-plural/#72af21dc5475
(2019/06/24)

"attached is the data vs attached are the data"

A complete search of the internet has found these results:
attached is the data is the most popular phrase on the web.

attached is the data

1,240 results on the web

attached are the data

103 results on the web
https://textranch.com/93434/attached-is-the-data/or/attached-are-the-data/
(2019/06/24)