Department of Computer Science and Engineering,
PES University, Bangalore, India

Lecture Notes
Problem Solving With C
UE24CS151B

Lecture #1
Prelude

By,

Prof. Sindhu R Pai,

Theory Anchor, Feb-May, 2025

Assistant Professor

Dept. of CSE, PESU

Many Thanks to

Dr. Shylaja S S (Director, CCBD and CDSAML Research Center, PES University)

Prof. Nitin V Poojari (Dean, Internal Quality Assurance Cell, PES University)

Unit #: 1

Unit Name: Problem Solving Fundamentals

Topic: Prelude

Course objectives: The objective(s) of this course is to make students

- Acquire knowledge on how to solve relevant and logical problems using computing Machine.
- Map algorithmic solutions to relevant features of C programming language constructs.
- Gain knowledge about C constructs and its associated ecosystem.
- Appreciate and gain knowledge about the issues with C Standards and it's respective behaviours.

Course outcomes: At the end of the course, the student will be able to:

- Understand and Apply algorithmic solutions to counting problems using appropriate C Constructs.
- Understand, Analyze and Apply sorting and Searching techniques.
- Understand, Analyze and Apply text processing and string manipulation methods using Arrays, Pointers and functions.
- Understand user defined type creation and implement the same using C structures, unions and other ways by reading and storing the data in secondary systems which are portable.

Sindhu R Pai

Theory Anchor, Feb - May, 2025

Dept. of CSE,

PES University

Prelude

Let us answer few questions before we start with C.

Q1. What is a Computer Programming Language (CPL)?

- Any set of rules that converts strings, or graphical program elements in the case of visual programming languages, to various kinds of **machine code output.**
- A **CPL** is an **artificial language** that can be used to control the behaviour of a machine, particularly a computer.
- **CPLs**, like human languages, are defined through the use of syntactic and semantic rules, to determine structure and meaning respectively.
- **CPLs** are used to implement **algorithms**.
- **CPLs allow us to give instructions to a computer** in a language the computer understands.
- **Formal computer language or constructed language** designed to communicate instructionsto a machine, particularly a computer (wiki definition).
- Can be used to create programs **to control the behaviour of a machine.**

Q2. Why CPL?

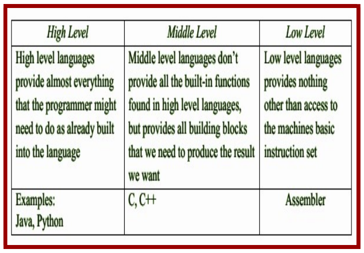
- Advance our ability to develop real algorithms.
- Majority of CPLs come with a lot of features for the Computer Programmers - CP.
- CPLs can be used in a proper way to get the best results.
- Improve Customization of our Current Coding.
- By using basic features of the existing CPL we can simplify things to program a better option to write resourceful codes.
- There is no compulsion of writing code in a specific way, but rather is the usage of features used and clarity of the concept.

Q3. Why so many Programming Languages?

To choose the right language for a given problem. Example Domains are listed – Web Browsers, Social Networks, Image Viewer, Facebook.com, Bing, Google.com, Games, Various Operating Systems.

Q4. What are the different Levels of Programming Language?

- **Low Level**
  - Binary codes which CPU executes
  - Programmer's responsibility is more
  - Machine Language
- **Middle Level**
  - Offers basic data structure and array definition, but the programmers shouldtake care of the operations
  - C and C++
- **High Level**
  - Programmer concentrates on the algorithm and programming itself
  - Java , Python, Pascal



High Level	*Middle Level*	*Low Level*
High level languages provide almost everything that the programmer might need to do as already built into the language	Middle level languages don't provide all the built-in functions found in high level languages, but provides all building blocks that we need to produce the result we want	Low level languages provides nothing other than access to the machines basic instruction set
Examples: Java, Python	C, C++	Assembler

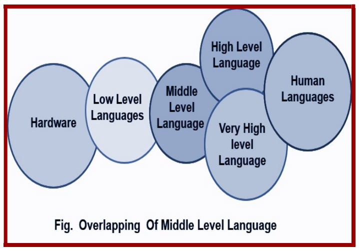


Fig. Overlapping Of Middle Level Language

Q5. What is the meaning of Paradigm?

A programming paradigm is a **style, or "way," of programming**. Some languages make it easy to write in some paradigms but not all of them.

Imperative: Programming with an explicit sequence of commands that update state – Example: python

Declarative: Programming by specifying the result you want, not how to get it Example: LISP, SQL

Structured: Programming with clean, goto-free, nested control structures

Example: C Language

Procedural: Imperative programming with procedure calls

Example: C Language.

Functional (Applicative): Programming with function calls that avoid any global state

Examples: Scheme, Haskell, Miranda and JavaScript.

Function-Level (Combinator): Programming with no variables at all

Examples: Scheme, Haskell, Miranda and JavaScript.

Object-Oriented: Programming by defining objects that send messages to each other. Objects have their own internal (encapsulated) state and public interfaces.

Class-based: Objects get state and behavior based on membership in a class.

Prototype-based: Objects get behavior from a prototype object.

Examples: Java, C++, Python

Event-Driven: Programming with emitters and listeners of asynchronous actions.

Flow-Driven: Programming processes communicating with each other over predefined channels.

Logic (Rule-based): Programming by specifying a set of facts and rules. An engine infers the answers to questions.

Constraint: Programming by specifying a set of constraints. An engine finds the values that meet the constraints.

Aspect-Oriented: Programming cross-cutting concerns applied transparently.

Reflective: Programming by manipulating the program elements themselves.

Array: Programming with powerful array operators that usually make loops unnecessary.

Q6: Which Languages are used while Developing Whatsapp? Think. !

Erlang, JqGrid, Libphonenum, LightOpenId, PHP5, Yaws and many more…

Q7. Why should one learn C? What are the advantages of 'C'? Is not 'C' an outdated language?

We have to fill our stomach every day 3 or 4 times so that our brain and body get enough energy to function. How about eating Vidyarthi Bhavan Dosa every day? What about Fridays when the eatery is closed? Why not buy Dosa batter from some nearby shop? Or do you prefer to make the batter yourself? Would you have time to do that? Would that depend on how deep your pockets are? Would you like to decrease your medical bills?

Every language has a philosophy. The language used by poets may not be suitable for conversation. Poets use ambiguity in meaning to their advantage, and some verses in Sanskrit have more than one meaning. But that will not be suitable for writing a technical report. The goal of 'C' is efficiency. The safety is in the hands of the programmer. 'C' does very little apart from what the programmer has asked for.

Example: When we index outside the bounds of a list in Python, we get an "index error" at runtime. To support this feature, Python runtime should know the current size of a list and should also check whether the index is valid each time we index on a list. You are all very good programmers, and I am sure you never get an index error. You get what you deserve. If you are lucky, the program crashes. Otherwise, something subtle may happen, which later may lead to catastrophic failures**.**

- C gives importance to efficiency
- C is not very safe; you can make your program safe
- C is not very strongly typed; mixing of types may not result in errors
- C is the language of choice for all hardware related softwares
- C is the language of choice for software's like operating system, compilers, linkers, loaders, device drivers.

Q8. Is 'C' not an old language?

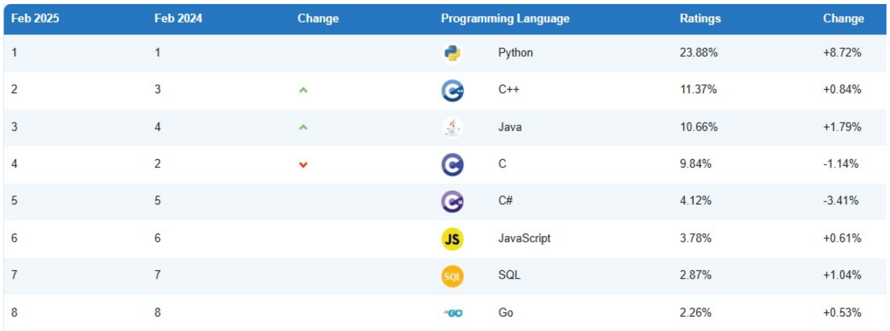
Yes and No.

It was designed by Dennis Ritchie –We use 'C' like languages and Unix like operating systems both have his contribution – in 70s. But the language has evolved over a period. The latest 'C' was revised in 2011.

There is one more reason to learn ‘C’. ‘C’ is the second most popular language as of now according to TIOBE index ratings. https://www.tiobe.com/tiobe-index/.

Q9. What is TIOBE Index?

- **The Importance Of Being Earnest” - TIOBE.**
- TIOBE is an indicator of the popularity of programming languages.
- The TIOBE index is updated once a month.
- The ratings are based on the number of skilled engineers world-wide, courses and third party vendors.
- Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Bing are used
- It is not about the best programming language or the language in which most lines of code have been written.



Feb 2025	Feb 2024	Change	Programming Language	Ratings	Change
1	1		Python	23.88%	+8.72%
2	3	▲	C++	11.37%	+0.84%
3	4	▲	Java	10.66%	+1.79%
4	2	▼	C	9.84%	-1.14%
5	5		C#	4.12%	-3.41%
6	6		JavaScript	3.78%	+0.61%
7	7		SQL	2.87%	+1.04%
8	8		Go	2.26%	+0.53%

Q10: What is the history of PLs?

- 1820-1850 England, Charles Babbage invented two mechanical Computational device i.e., Analytical Engine and Difference Engine
- In 1942, United States, ENIAC used electrical signals instead of physical motion
- In 1945, Von Newman developed two concepts: Shared program technique and Conditional control transfer
- In 1949, Short code appeared

- In 1951, Grace Hopper wrote first compiler, A-0
- Fortran: 1957, John Backus designed.
- Lisp, Algol- 1958
- Cobol: 1959
- Pascal: 1968, Niklaus Wirth
- **C: 1972, D Ritchie**
- C++: 1983, Bjarne Stroustrup, Compile time type checking, templates are used.
- Java: 1995, J. Gosling, Rich set of APIs and portable across platform through the use of JVM
- **Development of C**
  - **Martin Richards, around 60's developed BCPL [Basic Combined Programming Language]**
  - **Enhanced by Ken Thompson and Introduced B language.**
  - **C is originally developed between 1969 and 1973 at Bell Labs by Dennis Ritchie and Kernighan. Closely tied to the development of the Unix operating system**
- **Standardized by the ANSI [American National Standards Institute] since 1989 and subsequently by ISO [International Organization for Standardization].**
  - **ANSI C, C89, C99, C11, C17, or C23**

Happy Coding!