Introduction to C Programming

Advanced C

Have you ever pondered how
- powerful it is?
- efficient it is?
- flexible it is?
- deep you can explore your system?

if [NO]
    Wait!! get some concepts right before you dive into it
else
    You shouldn't be here!!

EMERTXE

Advanced C

Introduction - Where is it used?

- System Software Development
- Embedded Software Development
- OS Kernel Development
- Firmware, Middle-ware and Driver Development
- File System Development

  And many more!!

Advanced C

Introduction - Language – What?

- A stylized communication technique
- Language has collection of words called “Vocabulary”
  - Rich vocabulary helps us to be more expressive
- Language has finite rules called “Grammar”
  - Grammar helps us to form infinite number of sentences
- The components of grammar :
  - The ***syntax*** governs the structure of sentences
  - The ***semantics*** governs the meanings of words and sentences

Advanced C

Introduction - Language – What?

- A stylized communication technique
- It has set of words called “keywords”
- Finite rules (Grammar) to form sentences (often called expressions)
  - Expressions govern the behavior of machine (often a computer)
- Like natural languages, programming languages too have :
  - Syntactic rules (to form expressions)
  - Semantic rules (to govern meaning of expressions)

Advanced C

Introduction - Brief History

- Prior to C, most of the computer languages (such as Algol) were academic oriented, unrealistic and were generally defined by committees.
- Since such languages were designed having application domain in mind, they could not take the advantages of the underlying hardware and if done, were not portable or efficient under other systems.
- It was thought that a high-level language could never achieve the efficiency of assembly language

Portable, efficient and easy to use language was a dream.

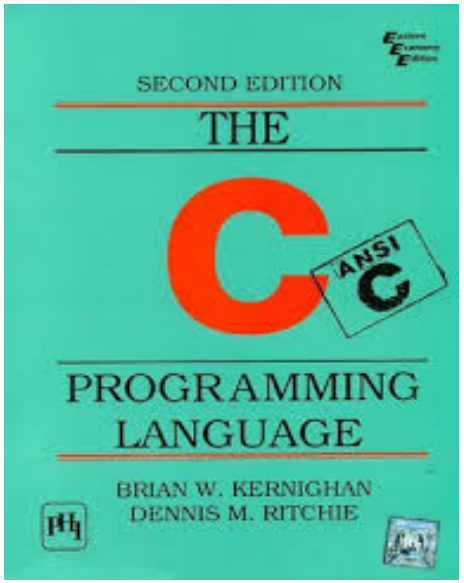
Advanced C

Introduction - Brief History

- It was a revolutionary language and shook the computer world with its might. With just 32 keywords, C established itself in a very wide base of applications.
- It has lineage starting from CPL, (Combined Programming Language) a never implemented language
- Martin Richards implemented BCPL as a modified version of CPL. Ken Thompson further refined BCPL to a language named as B
- Later Dennis M. Ritchie added types to B and created a language, what we have as C, for rewriting the UNIX operating system

Advanced C

Introduction - Standard



- “The C programming language” book served as a primary reference for C programmers and implementers alike for nearly a decade
- However it didn’t define C perfectly and there were many ambiguous parts in the language
- As far as the library was concerned, only the C implementation in UNIX was close to the ’standard’

- So many dialects existed for C and it was the time the language has to be standardized and it was done in 1989 with ANSI C standard
- Nearly after a decade another standard, C9X, for C is available that provides many significant improvements over the previous 1989 ANSI C standard

Advanced C

Introduction - Important Characteristics

- Considered as a middle level language
- Can be considered as a pragmatic language
- It is indented to be used by advanced programmers, for serious use, and not for novices and thus qualify less as an academic language for learning
- Gives importance to compact code
- It is widely available in various platforms from mainframes to palmtops and is known for its wide availability

Advanced C

Introduction - Important Characteristics

- It is a general-purpose language, even though it is applied and used effectively in various specific domains
- It is a free-formatted language (and not a strongly-typed language)
- Efficiency and portability are the important considerations
- Library facilities play an important role

Advanced C

Introduction - Keywords

- In programming, a keyword is a word that is reserved by a program because the word has a special meaning
- Keywords can be commands or parameters
- Every programming language has a set of keywords that cannot be used as variable names
- Keywords are sometimes called reserved names

Advanced C

Introduction - Keywords - Categories

Type	Keyword
Data Types	char int float double
Modifiers	signed unsigned short long
Qualifiers	const volatile
Loops	for while do
Jump	goto break continue

Type	Keyword
Decision	if else switch case default
Storage Class	auto register static extern
Derived	struct unions
User defined	enums typedefs
Others	void return sizeof

Advanced C

Introduction - Typical C Code Contents

Documentation

Preprocessor Statements

Global Declaration

The Main Code:

Local Declarations
Program Statements
Function Calls

One or many Function(s):

The function body

- A typical code might contain the blocks shown on left side
- It is generally recommended to practice writing codes with all the blocks

Advanced C

Introduction - Anatomy of a Simple C Code

Advanced C

Introduction - Compilation

- Assuming your code is ready, use the following commands to compile the code
- On command prompt, type

  $ gcc <file_name>.c

- This will generate a executable named a.out
- But it is recommended that you follow proper conversion even while generating your code, so you could use

  $ gcc <file_name>.c -o <file_name>

- This will generate a executable named <file_name>

Advanced C

Introduction - Execution

- To execute your code you shall try

  ```
  $ ./a.out
  ```

- If you have named you output file as your <file_name> then

  ```
  $ ./<file_name>
  ```

- This should the expected result on your system