UNIT – 1
Problem solving with computer (Part II)

C PROGRAMMING (CSC115)
BSC CSIT FIRST SEMESTER (TU)

PREPARED BY:

DABBAL SINGH MAHARA

ASCOL (2025)

INTRODUCTION TO C

- C programming is a **general-purpose, procedural programming language** that was developed by **Dennis Ritchie** in 1972 at Bell Labs.
- It is considered the "mother" of many modern languages, including C++, C#, Java, and Python, due to their borrowed syntax and structure.
- It builds a foundation for learning **C++, Java, Python** and understanding **fundamentals of programming** (loops, conditionals, data types).
- Used in **system software, embedded systems, compilers, and OS development**.

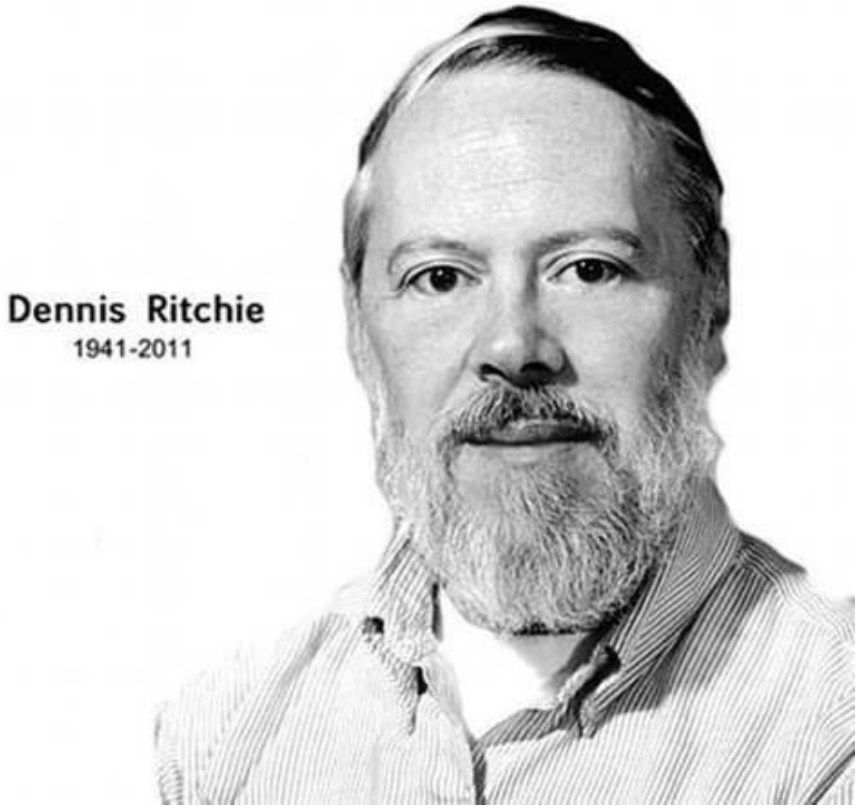
C is	Description
• **Procedure-Oriented**	Programs are divided into **functions**; focuses on step-by-step procedures to solve problems.
• **Structured Language**	Supports **sequence, selection, and iteration**; code can be organized into modular blocks (functions).
• **Free-Form Language**	**Spaces, line breaks, and indentation** do not affect program execution; flexible formatting.
• **Statically Typed**	Variables must be **declared with type** before use; type errors are caught during compilation.
• **Portable Language**	C programs can run on **different platforms** with minimal modification.
• **Compiled Language**	Requires a **compiler** to convert code into machine language before execution.
• **General-Purpose Language**	Suitable for a variety of applications: **system software, OS, embedded systems, and applications**.

History of C language

- C is a by-product of UNIX OS.
- UNIX was written in assembly language by **Ken Thompson** for DEC PDP-7 at Bell Lab. But it was very painful to debug and hard to enhance.
- So, Thompson himself developed a small language named B for further development of UNIX. B was based on BCPL, a system programming language during mid 1960s. BCPL was based on Algol 60 one of the influential language of 1960.
- In 1970 Bell Lab acquired Dec PDP-11 for Unix project and UNIX rewritten in B to run on this new machine. In 1971 it was realized that B was not suitable language for PDP-11.

History of C language

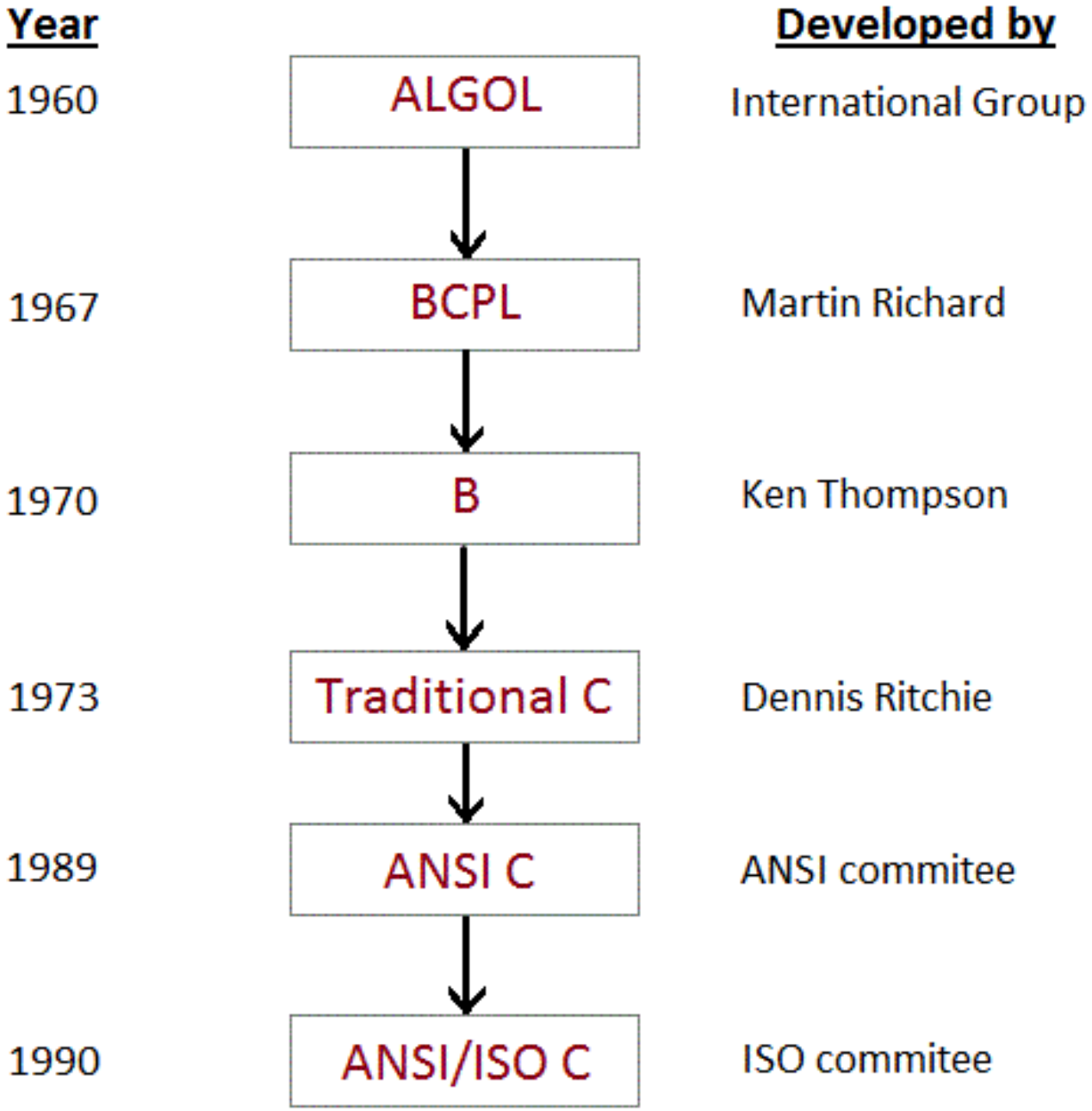
- So, Ritchie began to develop extended version of B. He called his new language as New B (NB) at first. But, it began diverge more from B, so he changed name to C.
- UNIX was rewritten in C in 1973.
- The switch to C provided an important benefit: **Portability.**
- By writing C compilers for other computers at Bell Lab the team could get UNIX running on those machines as well.



History of C language

- In 1978, Brian Kernighan and Dennis Ritchie produced the first publicly available description of C, as a book, “ **The C Programming Language**”, now known as the **K&R standard**.
- In 1989, the C standard was ratified as "Programming Language C". This version of the language is often referred to as ANSI C, Standard C, or sometimes C89.
- In 1990, the ANSI C standard (with formatting changes) was adopted by the International Organization for Standardization (ISO) which is sometimes called C90. Therefore, the terms "C89" and "C90" refer to the same programming language.

History of C language



Basic Features of C Language

1. **Simple :** C is a simple language in the sense that it provides a structured approach (to break the problem into parts), the rich set of library functions, data types, etc.
2. **Machine Independent or Portable:** Unlike assembly language, c programs can be executed on different machines with some machine specific changes. Therefore, C is a machine independent language.
3. **Mid-level programming language:** Although, C is intended to do low-level programming. It is used to develop system applications such as kernel, driver, etc. It also supports the features of a high-level language. That is why it is known as mid-level language.
4. **Structured programming language:** C is a structured programming language in the sense that we can break the program into parts using functions. So, it is easy to understand and modify. Functions also provide code reusability.
5. **Rich Library:** C provides a lot of inbuilt functions that make the development fast.

Basic Features of C Language

6. **Memory Management:** It supports the feature of dynamic memory allocation. In C language, we can free the allocated memory at any time by calling the free() function.
7. **Speed:** The compilation and execution time of C language is fast since there are lesser inbuilt functions and hence the lesser overhead.
8. **Pointer:** C provides the feature of pointers. We can directly interact with the memory by using the pointers. We can use pointers for memory, structures, functions, array, etc.
9. **Recursion:** In C, we can call the function within the function. It provides code reusability for every function. Recursion enables us to use the approach of backtracking.

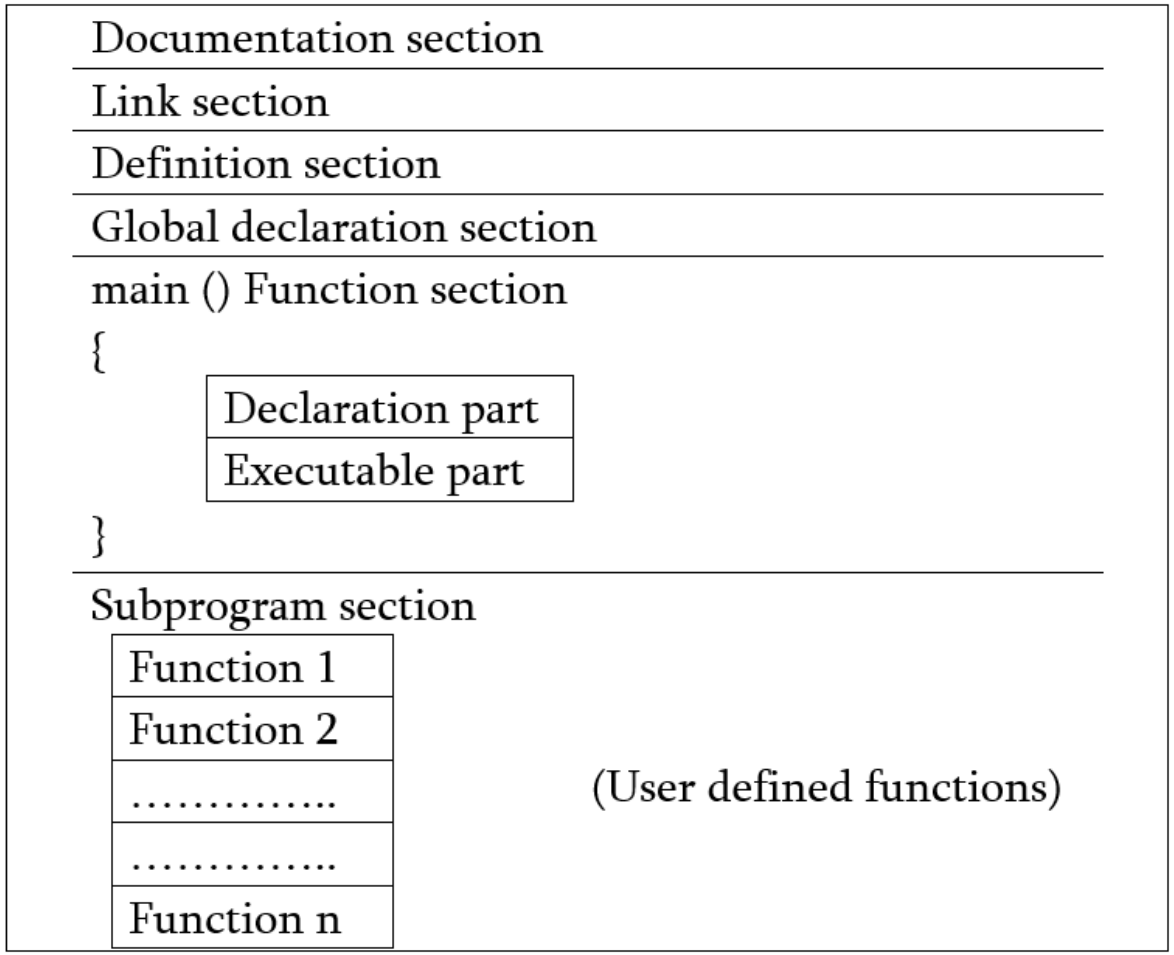
Key Applications

- 'C' language is widely used in embedded systems.
- Most of the applications by Adobe are developed using 'C' programming language.
- It is used to develop databases. Oracle, MySQL are popular database software built using 'C'.
- It is used in developing an operating system. Operating systems such as Apple's OS X, Microsoft's Windows, and Symbian are developed using 'C' language.
- It is used for compiler production.
- It is widely used in IOT applications.

The limitations of C

- Difficult to debug.
- C compilers can only identify syntax errors and are incapable of handling exceptions (run-time errors).
- C provides no data protection.
- It does not provide strict data type checking (for example an integer value can be passed for floating datatype).
- C Programming Language doesn't support Object Oriented Programming(OOP) features like Inheritance, Encapsulation, Polymorphism etc.

Structure of a C program



Structure of a C program

Documentation	Consists of comments, some description of the program, programmer name and any other useful points that can be referenced later.
Link	Provides instruction to the compiler to link function from the library function.
Definition	Consists of symbolic constants.

Structure of a C program

Global declaration	Consists of function declaration and global variables.

main( )
{
 	
}
	Every C program must have a main() function which is the starting point of the program execution.

Subprograms	User defined functions.

First C Program

```
#include<stdio.h>

    int main()
    {
        printf("Hello World !");
        return 0;
    }
```

Structure of a C program: Example

```
/*
 * Program: circle.c
 * author: Dabbal Mahara
 * date: 2021-08-24
 * description: program to find the area of a circle
 *   using the radius r
 */

#include <stdio.h>
#define PI 3.1416
float area(float r);
```

```
int main(void)
{
  float r = 10;
  printf("Area: %.2f", area(r));
  return 0;
}

float area(float r) {
    return PI * r * r;
}
```

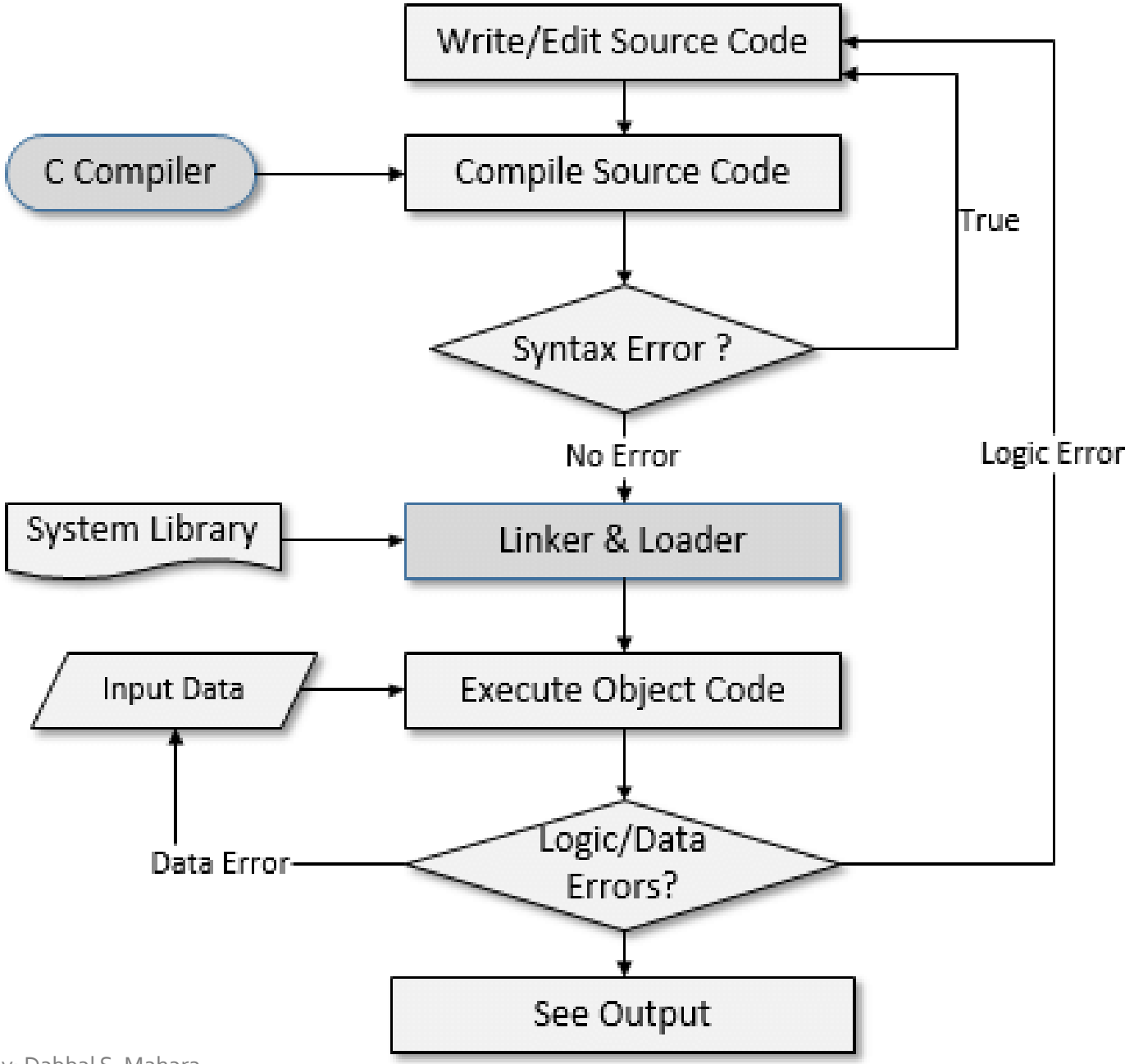
Points to be noted…

- Do not forget to use **pre-processor** at the beginning of the program.
- The program **must have the main function** with a return type or not return type.
- The program **must have the block** for defining the variables and writing code.
- C is **case sensitive** programming language so you do have to care about the variables, keywords and cases.
- Most of the time we use lower case for writing the codes in C, If you are using any other cases, you must have to remember.
- Statement **ends with semicolon** otherwise that will return a syntax error.
- **Use indentation** for better understanding the code and to make more readable and user-friendly.

The Steps for Compiling and executing C Program

- **Step1-**
  - Use a text editor **to write, edit or correct** source code into C source code files having the ".c" file extension.
  - A text editor is usually used to enter the C program into a file.
- **Step2-**
  - After the program has been written the next step is **compilation process** to translate the program using C compiler.
  - If the compiler doesn't find any syntax errors in the program source code, it produces an object file. The compiler produces object files with an ".obj" file extension and the same name as the source code.
  - If compiler finds any errors in your program, it reports them and you need to return to Step 1 to make corrections in your source code.

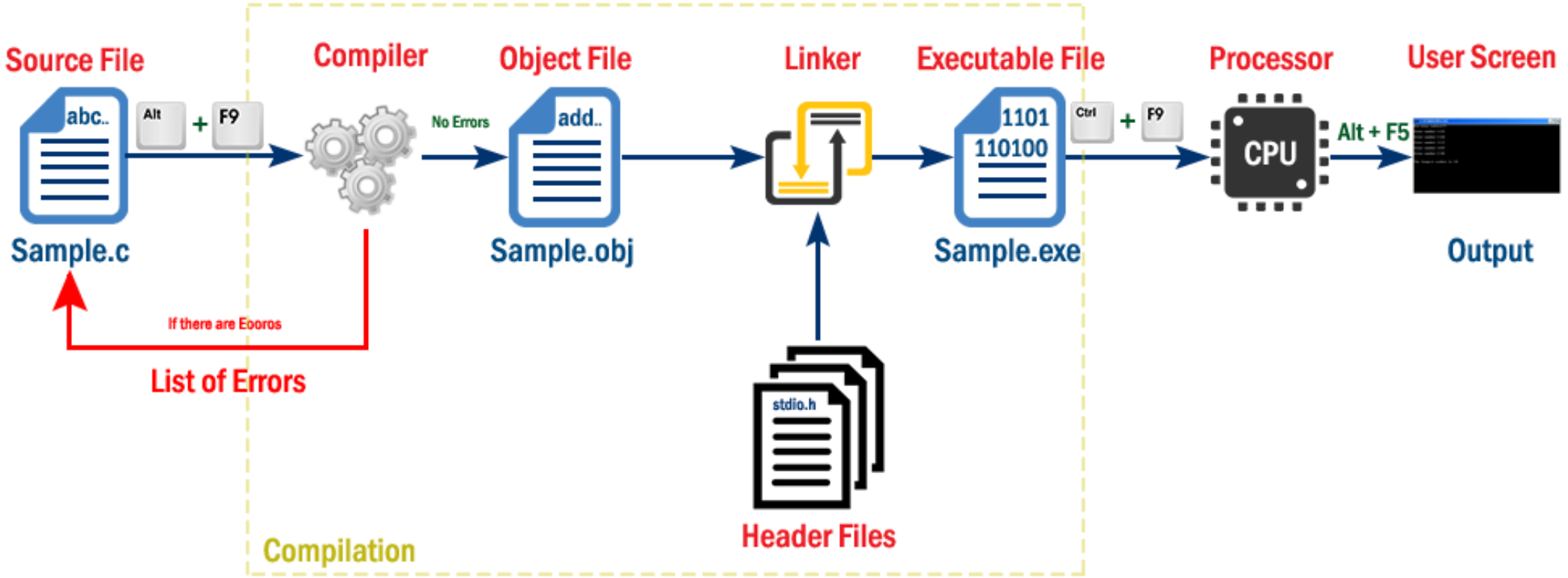
Process of compiling and running a C program



The Steps for Compiling and executing C Program

- **Step 3:**
  - After the program has been translated into object code, it is ready to link the program using the linker.
  - If no errors occur, the linker produces an executable program located in a disk file with an ".exe" file extension with the same name as the object file.
- Step 4:
  - Run the executable program and test whether it works properly as per the user requirements.
  - If the program does not produce the desired results, it is necessary to go back, start again with Step 1 and reanalyze the program's logic and make modifications and additions to your source code.

The Steps for Compiling and executing C Program



Integrated Development Environment (IDE)

- An IDE is a software that helps programmers write, run, and debug programs in one place.
- **Main Features included in IDE:**
  - **Code Editor** – To write and edit code.
  - **Compiler/Interpreter** – To convert code into executable form.
  - **Debugger** – To detect and fix errors.
  - **Build Tools** – To automate compiling and linking.
  - **Project Management** – To organize program files and resources.
  - **Syntax Highlighting** – Makes code easier to read.
  - **Code Completion** – Suggests code while typing.
- **Examples:** Visual Studio Code, Code::Blocks, Eclipse, NetBeans.

Assignment #2

1. Discuss the history of C briefly.
2. List the features of C.
3. Explain compilation and execution of C program with flowchart.
4. What are the different types of programming errors?
5. Explain the structure of C program with an example.


Thank
you!