

# INTRODUCTION TO PROGRAMMING

**By:**

**Mir Faraz Ali**

**Programming Fundamentals**

**Week 02,03**

---






# Recommended Readings

Chapter No. 1 and 2 of C++ Programming from Problem Analysis to Program Design written by DS Malik





# Topics to be Covered

- Recap of the previous lecture
  - Using IDE for writing the programs
  - Writing First program of C++
  - Examining First program of C++
  - How C++programs are processed and executed
  - Escape Sequences ('\n' and '\t')
  - Comments
  - Single Line Comments
  - Multiple Line Comments
  - Variables and Datatypes
  - Keywords
  - Countable Datatypes
  - Recap of the lecture
- 



# First Program of C++

- `#include <iostream>`
- `using namespace std ;`
- `int main ()`
- `{`
- `cout <<"Hello World... \n " ;`
- `system ("PAUSE") ;`
- `return 0 ;`
- `}`





# Compile and Execute

- For compilation press F9
- For compilation as well as execution press F10
- Compilation and Execution F11
- For saving the program go to file and click on save.
- By default file is saved with the extension of .CPP.





# Explanation of First Program

- #: It is a preprocessor directive
- include: It is a name of directory
- iostream: Built-in file
- using : Keyword
- Namespace: For directory
- Std: library file
- int: data type
- main (): Entry point for every C++ / C program
- { // Open curly bracket of the main function
- cout: output stream
- <<: Operator
- system ("PAUSE"): Built-in Function
- }// Closing curly brackets





# Preprocessor Directives



- Preprocessor commands are processed before the program goes through the compiler
- Many functions and symbols needed to run a C++ program are provided as collection of libraries
- Every library has a name and is referred to by a header file
- Preprocessor directives are commands supplied to the preprocessor program
- All preprocessor commands begin with #
- No semicolon at the end of these commands





# Preprocessor Directives

- Preprocessor commands are processed before the program goes through the compiler
- Many functions and symbols needed to run a C++ program are provided as collection of libraries
- Every library has a name and is referred to by a header file
- Preprocessor directives are commands supplied to the preprocessor program
- All preprocessor commands begin with #
- No semicolon at the end of these commands





# Processing of C++ Program

- To execute a C++ program:
- Use an editor to create a source program in C++
- Preprocessor directives begin with # and are processed by the preprocessor
- Use the compiler to:
- Check that the program obeys the language rules
- Translate into machine language (object program)





# Processing of C++ Program

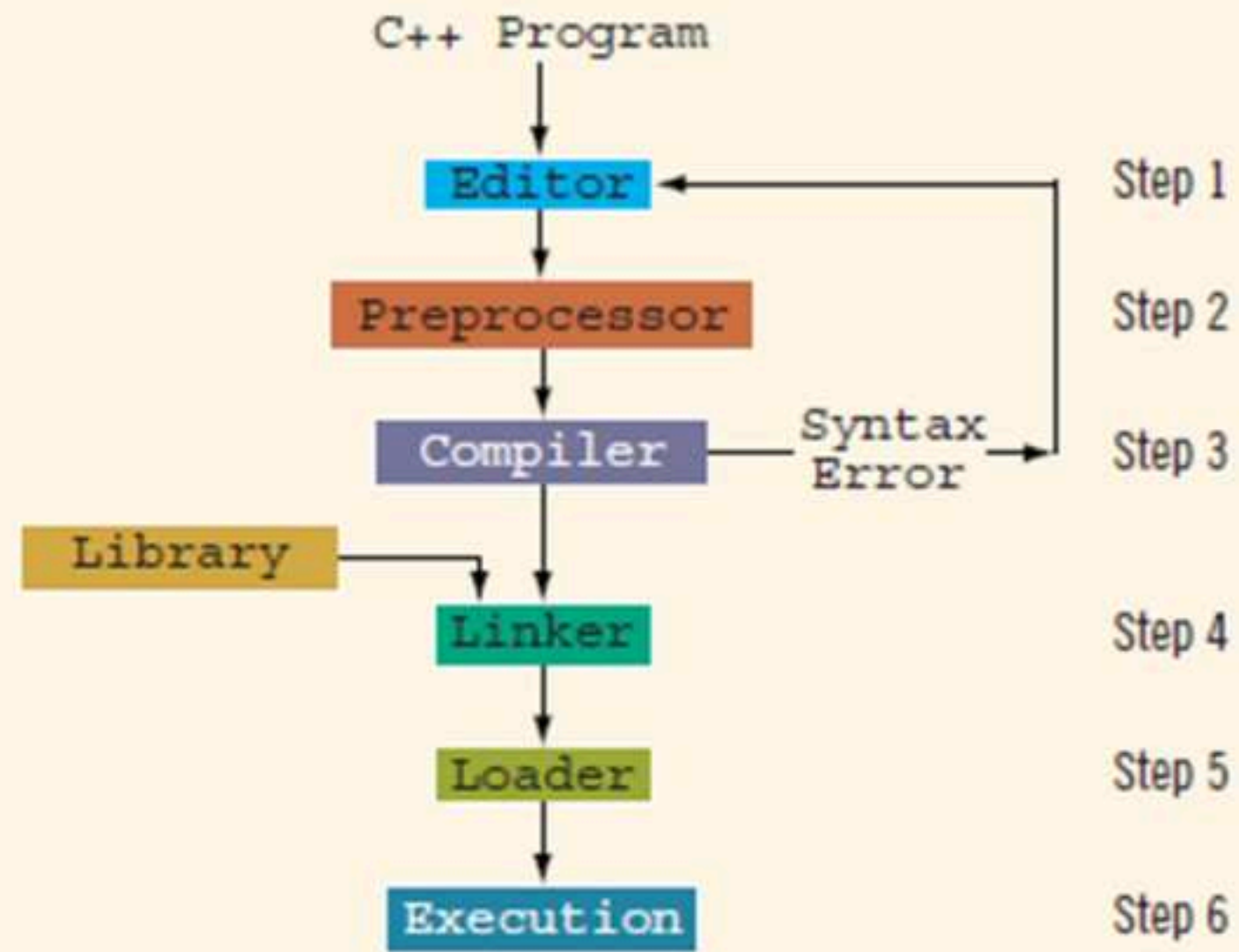
- To execute a C++ program (cont'd.):
- Linker:
- Combines object program with other programs provided by the SDK to create executable code
- Library: contains prewritten code you can use
- Loader:
- Loads executable program into main memory
- Execution
- The last step is to execute the program
- Some IDEs do all this with a Build or Rebuild

command



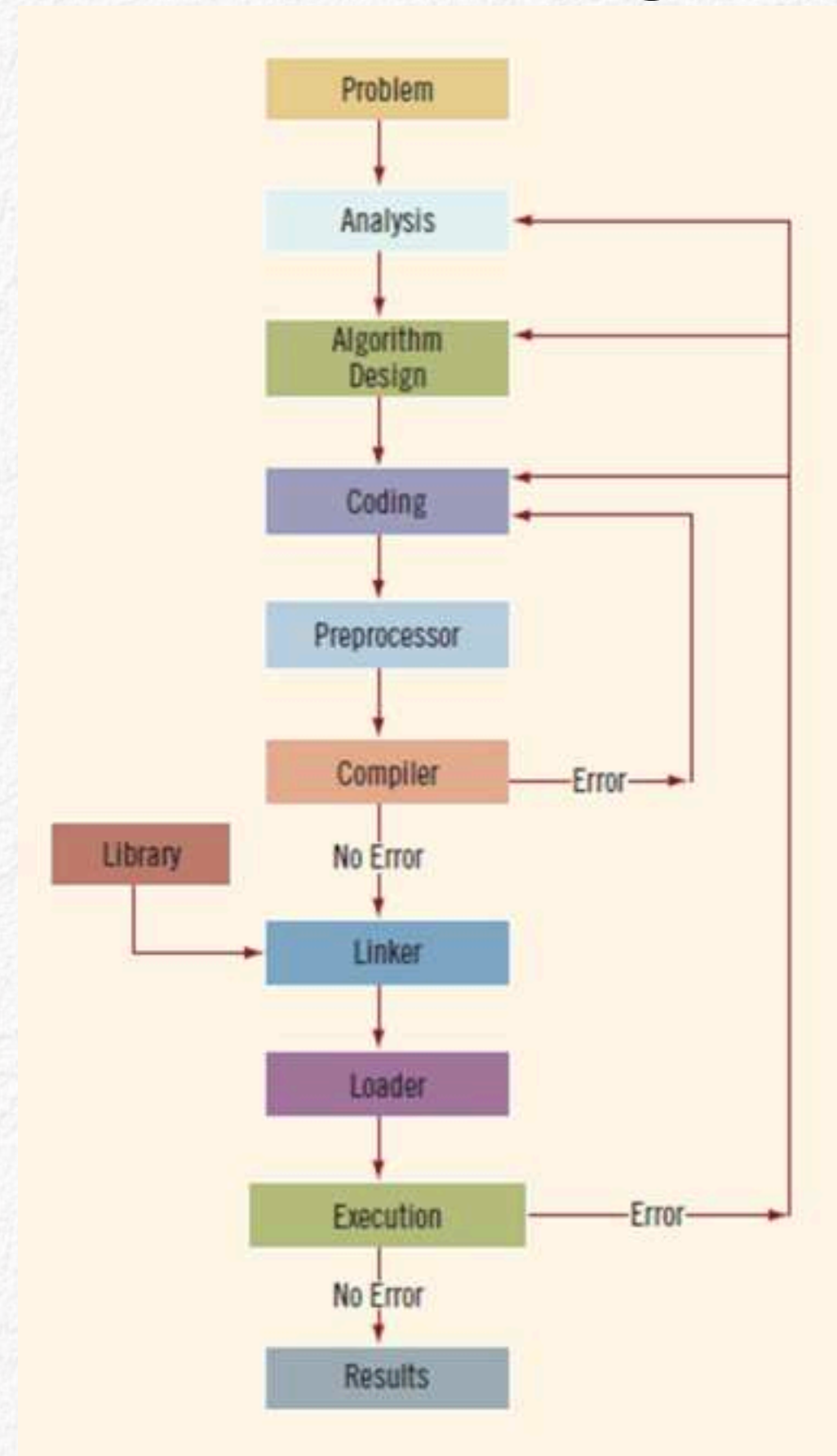


# Processing of C++ Program





# Processing of C++ Program





# Escape Sequences

- Certain special characters are represented as escape sequences.
- An escape sequence begins with a \ (backslash) followed by an alphanumeric character.
- List of Escape Sequence characters
  - \n New Line
  - \t Horizontal tab
  - \" Double quotes
  - \' Single quote
  - \v Vertical tab
  - \\ Back Slash
  - \? Question Mark





# COMMENTS



- Comments can be used to explain C++ code, and to make it more readable.
- Comments can be singled-lined or multi-lined.
- Single-line comments start with two forward slashes (`//`).
- Multi-line comments start with `/*` and ends with `*/`.





# END OF CLASS

---

