

Module 2 – Introduction to Programming

LAB EXERCISE

1. Overview of C Programming

Research and provide three real-world applications where C programming is extensively used, such as in embedded systems, operating systems, or game development.

1. Operating Systems

- Use: C is the backbone of many operating systems.
- Examples:
 - Linux Kernel is written mostly in C.
 - Windows and macOS also use C for core components.
- Why C?
 - Fast and close to hardware.
 - Offers fine control over memory and performance.

2. Embedded Systems

- Use: C is widely used to program microcontrollers and hardware devices.
- Examples:
 - Home appliances (washing machines, microwaves).
 - Medical devices, automotive control systems, smartwatches.
- Why C?
 - Small memory footprint.
 - Easy to interact with hardware directly.

3. Game Development (System-Level)

- Use: C is used to develop game engines or performance-critical parts of games.
- Examples:

- Doom, Quake, and early Unreal Engine used C.
 - Graphics drivers and audio engines.
- Why C?
 - High performance.
 - Low-level control helps in optimizing speed and graphics rendering.

2. Setting Up Environmen

Install a C compiler on your system and configure the IDE. Write your first program to print "Hello, World!" and run it.

➤ Step-by-Step to Run First C Program

1. Install Compiler

- Windows: Install Code::Blocks (includes compiler).
- Linux: Run
`sudo apt install build-essential`
- Mac: Run
`xcode-select --install`

2. Open IDE or Editor

- Use Code::Blocks or VS Code.

3. Write This Code:

```
c

#include <stdio.h>

int main() {

    printf("Hello, World!");

}
```

4. Run the Program

- In Code::Blocks → Press F9
- In terminal (VS Code or Linux):

```
bash  
gcc hello.c -o hello  
./hello
```

Output: Hello, World!