Linker and Loader

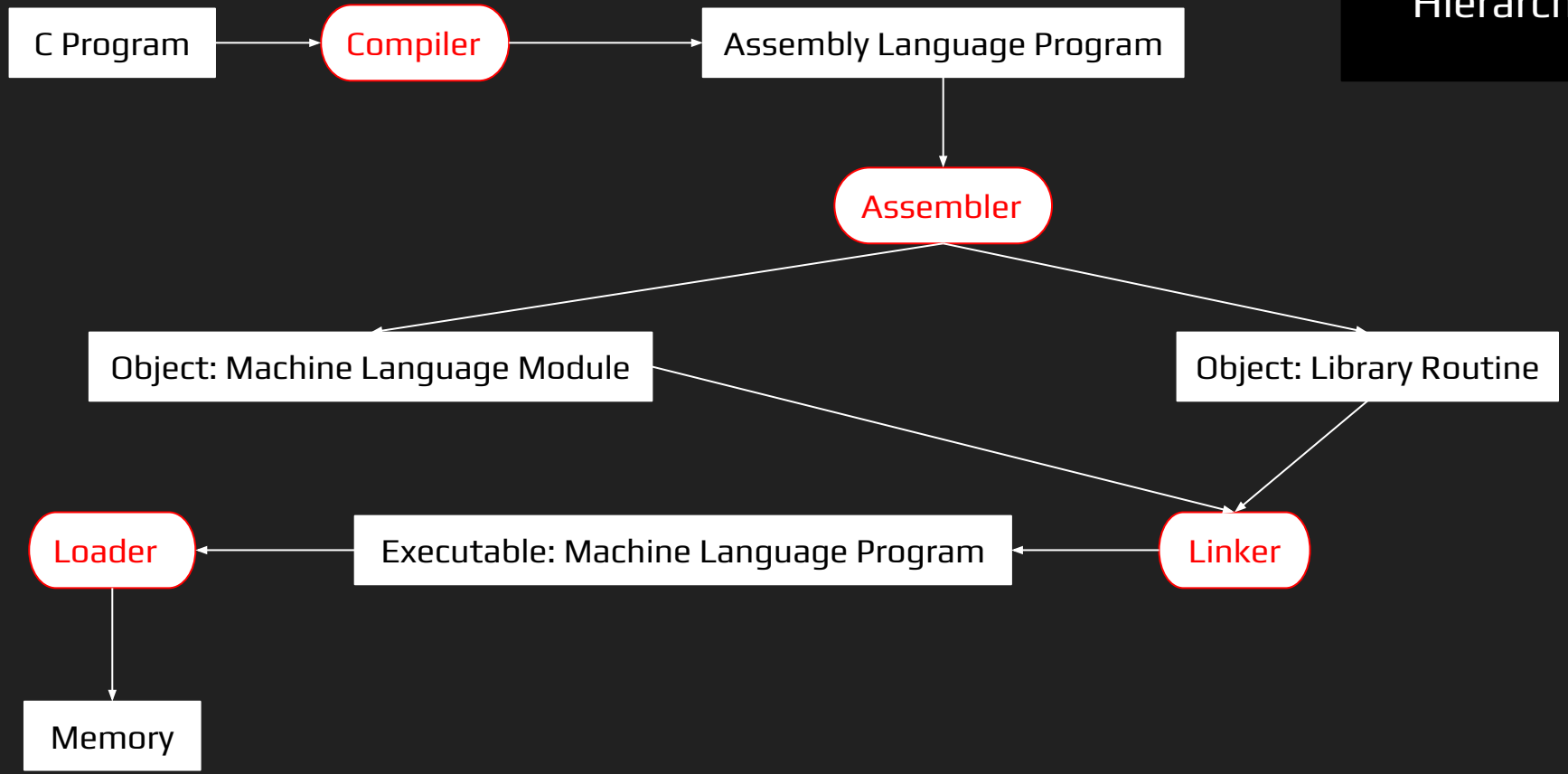
Presentation by:

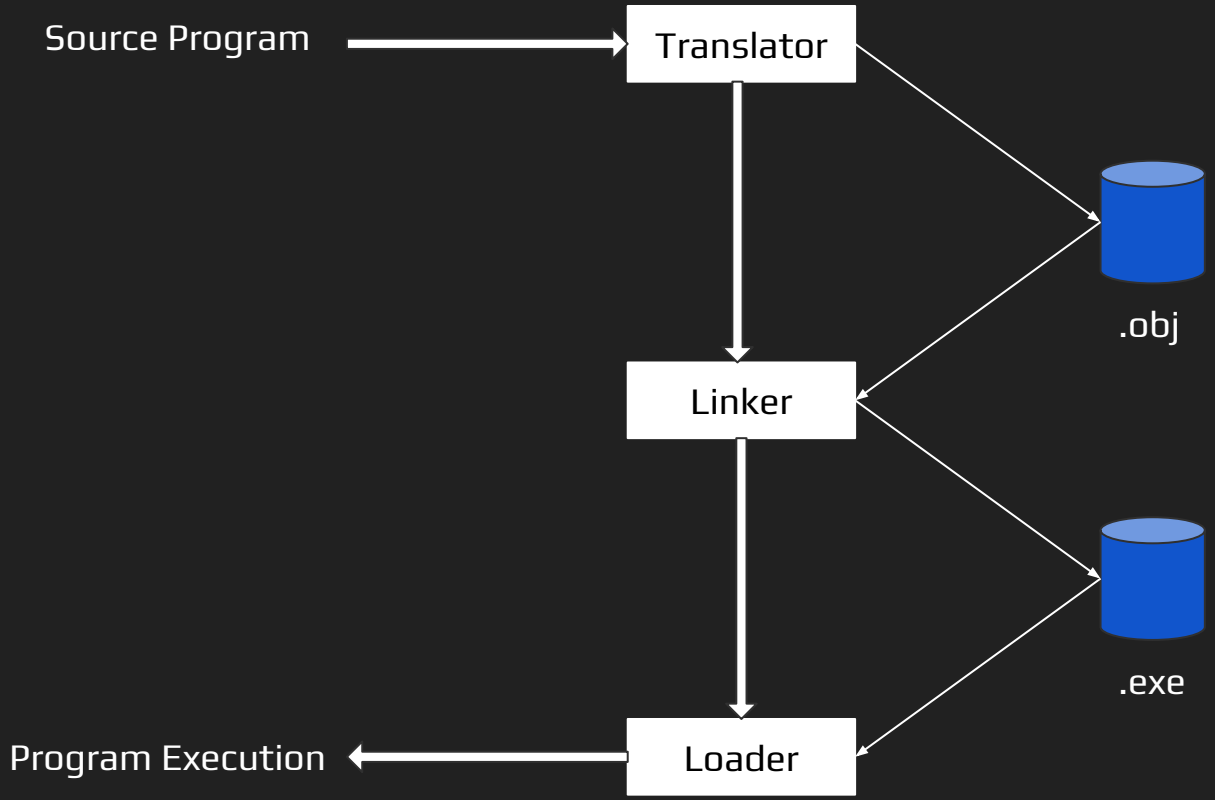
AMEY THAKUR	TE COMPS B-50	TU3F1819127
HASAN RIZVI	TE COMPS B-51	TU3F1819130
MEGA SATISH	TE COMPS B-58	TU3F1819139

Translation Hierarchy

- ➔ Compiler: Translates high-level language program into assembly language.
- ➔ Assembler: Converts assembly language programs into object files.
- ➔ Object files contain a combination of machine instructions, data, and information needed to place instructions properly in memory.

Translation Hierarchy





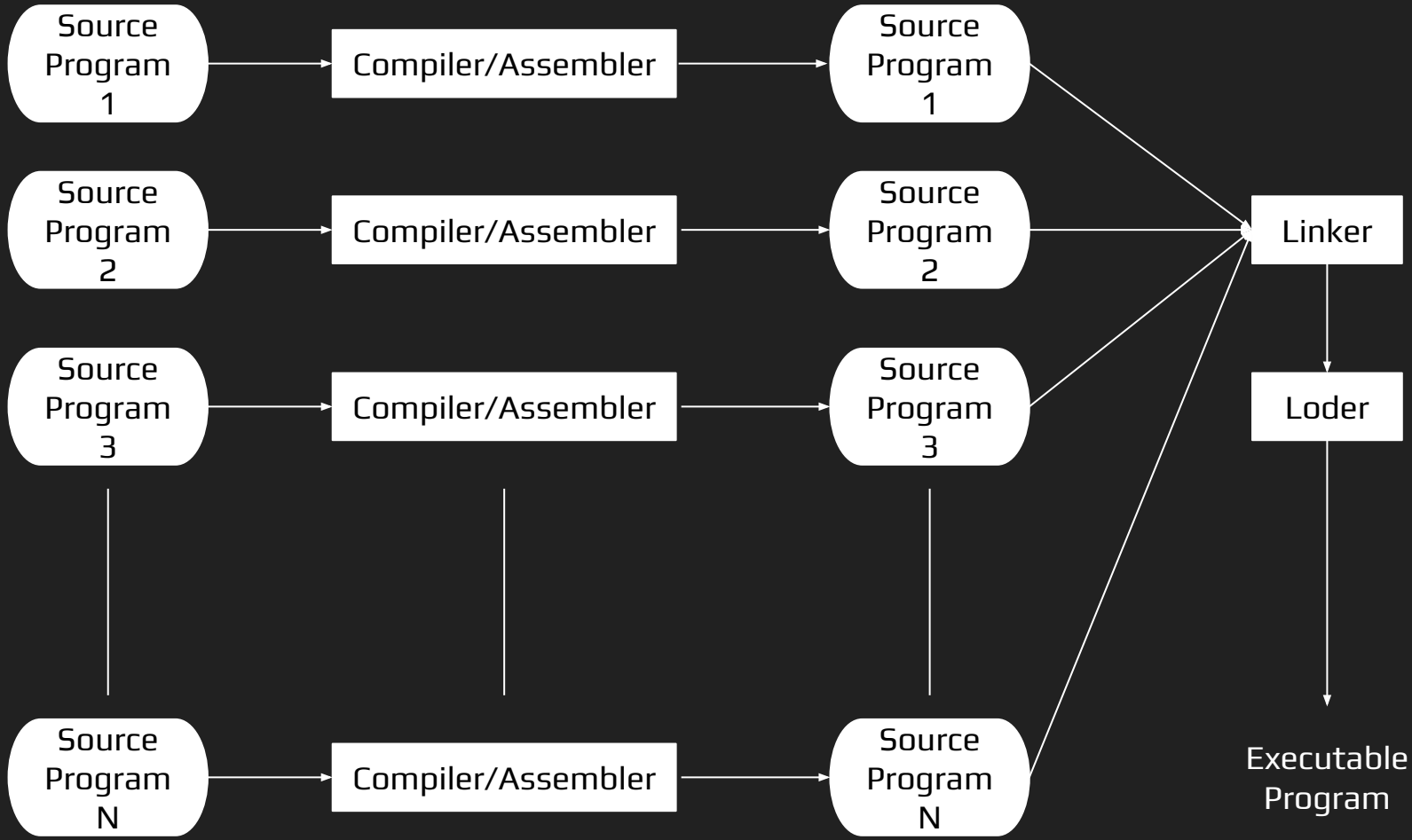
Flow of Execution

Linker

- ➔ Linker is a computer program that links and merges various object files together in order to make an executable file.

- ➔ All these files might have been compiled by separate assembler.

- ➔ The major task of a linker is to search and locate referenced module/routines in a program and to determine the memory location where these codes will be loaded, making the program instruction to have absolute references.


Source Program 1
Compiler/Assembler
Source Program 1
Source Program 2
Compiler/Assembler
Source Program 2
Source Program 3
Compiler/Assembler
Source Program 3
Source Program N
Compiler/Assembler
Source Program N
Linker
Loder
Executable Program

Loaders

- ➔ Loader is a program which accepts the input as linked modules and loads them into main memory for execution by the computer.
- ➔ Loaders load or copies program from secondary storage to main memory for execution.
- ➔ Loaders can also replace virtual addresses with physical addresses.

Functions Of Loader

- ➔ Allocation: Allocates space in memory for the program.
- ➔ Linking: Symbol Resolution between object modules.
- ➔ Relocation: Adjust addresses dependent locations of address constants i.e. assign load addresses to different parts of a program.
- ➔ Loading: Store machine instructions and data into memory.

Symbol Resolution

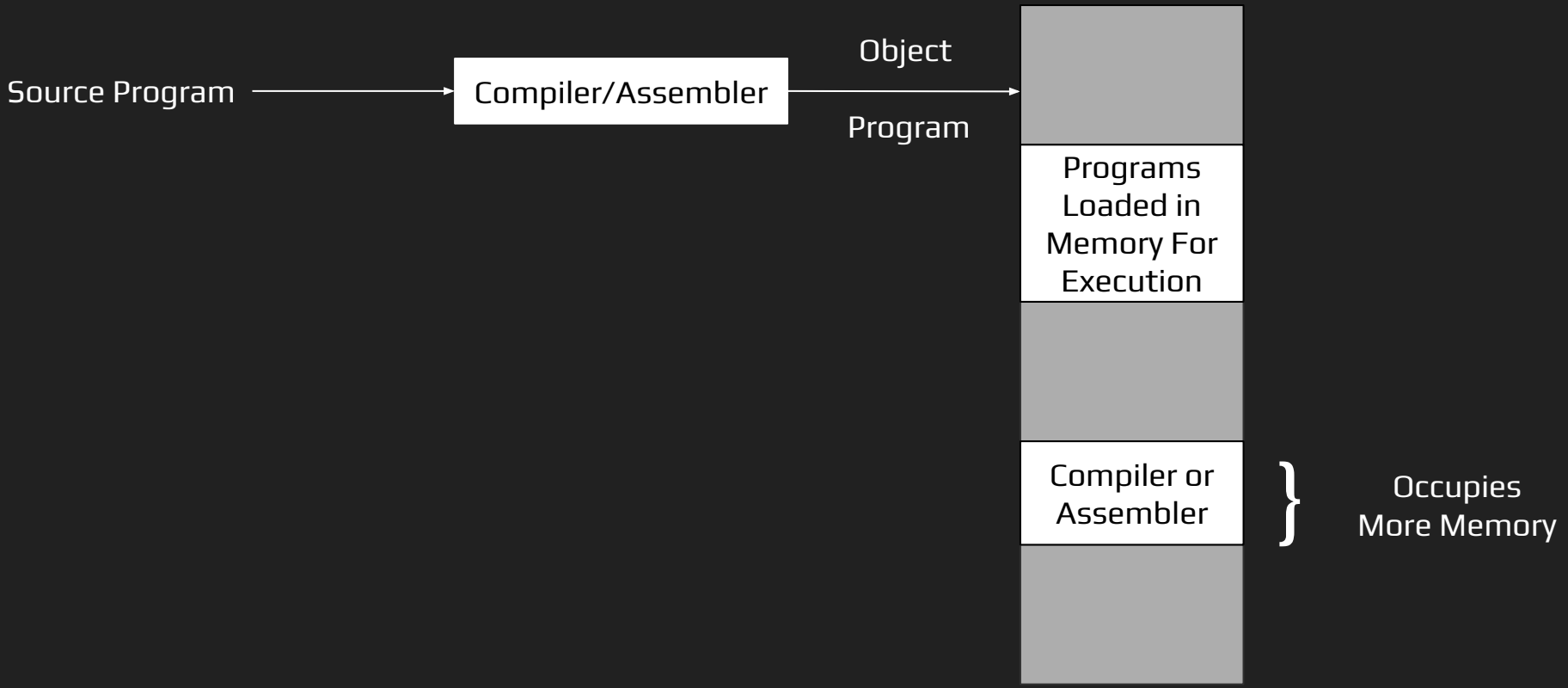
Program - 1

```
#include<stdio.h>
extern void show();
.
.
void  main()
{
        .
        .
        show();
        .
        .
}
void display()
{
        .
        .
}
```

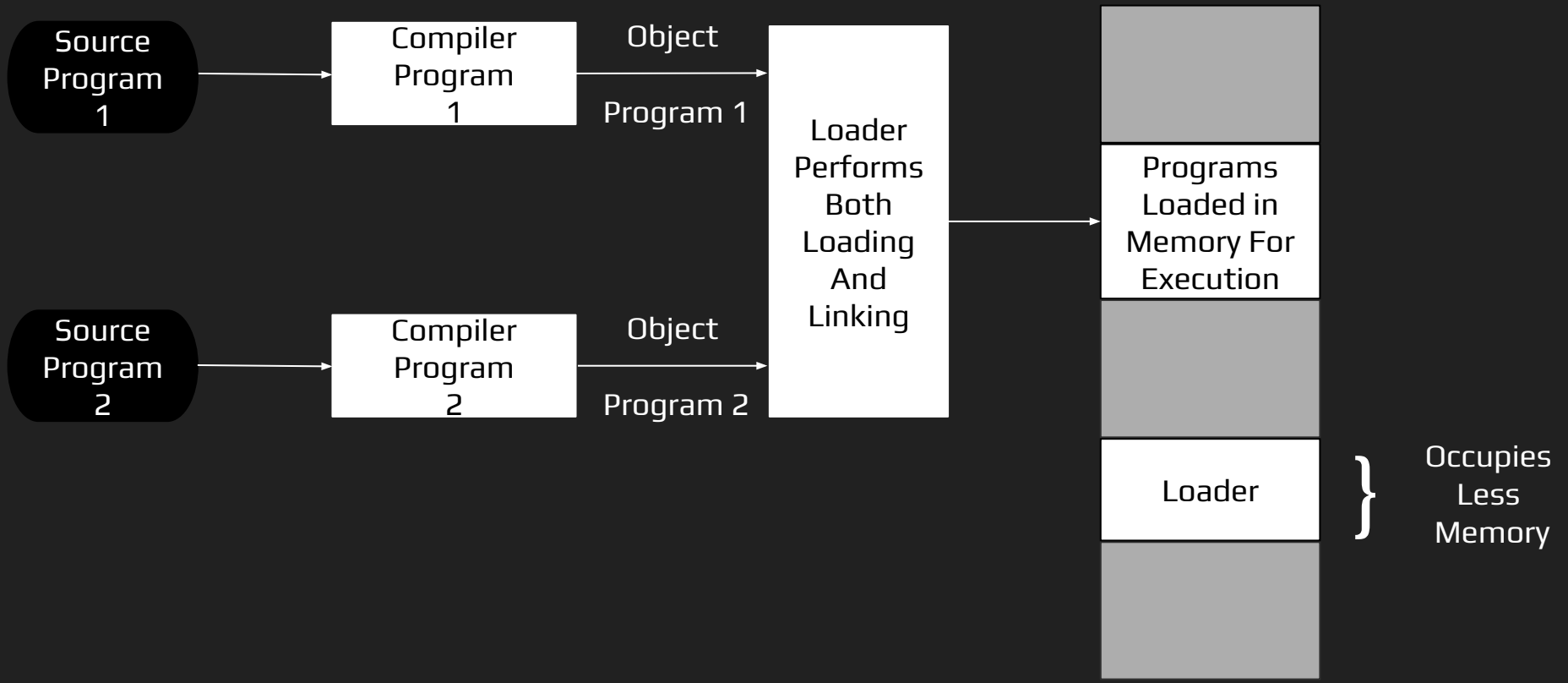
Program - 2

```
#include<stdio.h>
extern void display();
.
.
void  show()
{
        .
        .
        display();
        .
        .
}
```

Loading Scheme
(Simple Compilation without loader - Compile & Go Loading Scheme)



Loading Scheme
(Compilation with Loader - General Loading Scheme)



Types of Loaders

1. Absolute Loaders:
   - ➔ It is a primitive type of loader which does only the loading.
   - ➔ It does not perform linking and program relocation.
   - ➔ The programmer has to give the address in memory explicitly where he/she wants to store the program.

2. Relocatable Loader:
   - ➔ A loader which also performs relocation with loading.
   - ➔ It is the responsibility of relocation loader to load each function or subprogram at non-overlapping addresses and to give each function a original load address.