1.2 Structured Programming

Introduction

- Structured programing is the set of design and implementation processes that yield well-structured programs.

Why we need structured programming

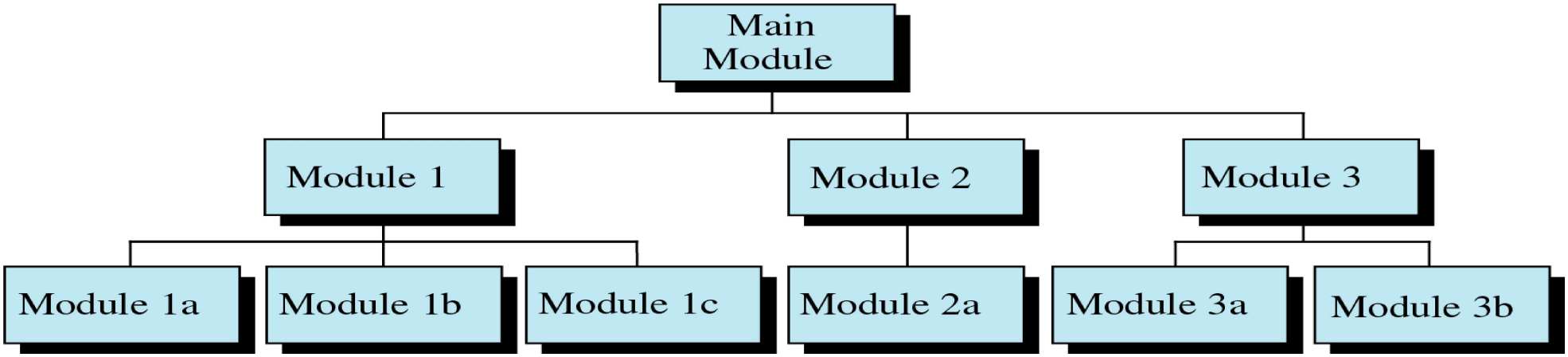
- It is a method of writing programs that are clear, easy to read, and easy to maintain.
- It uses control structures like sequence, selection, and repetition.
- Programs are divided into smaller modules or functions for better organization.

Basic Core Principles of Structured Programming:

- Top-down design – break problems into smaller sub problems.
- Stepwise refinement – solve in progressive detail.
- Three basic control structures:
  - ❑ Sequence
  - ❑ Selection (Choice)
  - ❑ Repetition (Loop)
- Importance of functions and Variable scope (Local & Global Variable)

Top-Down Design

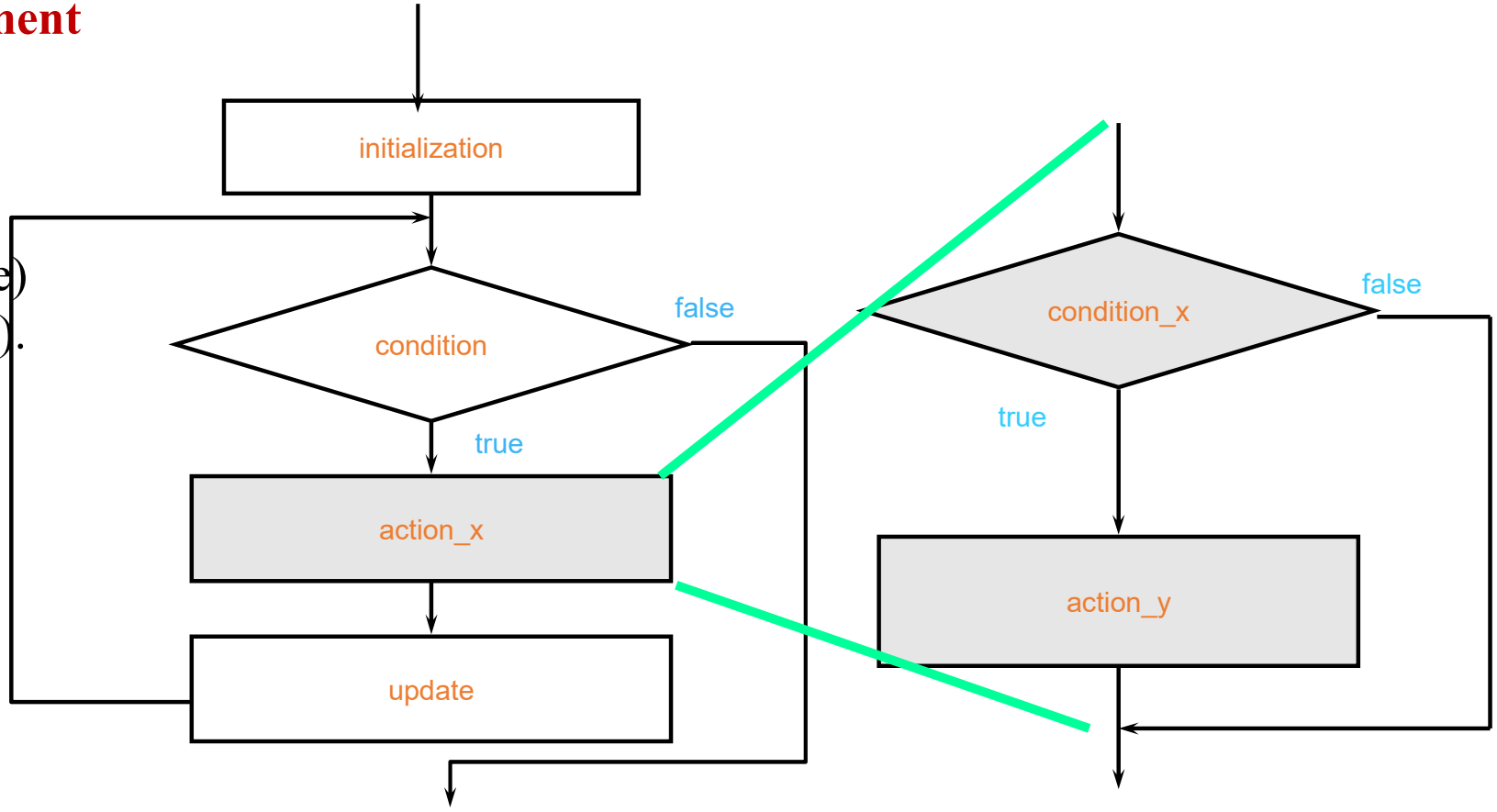
- A program is divided into a main module and its related modules.
- Each module is in turn divided into submodules until the resulting modules are understood without further division.



Stepwise Refinement

Any action can be another:

- Sequence,
- Selection (Choice)
- Repetition (Loop).

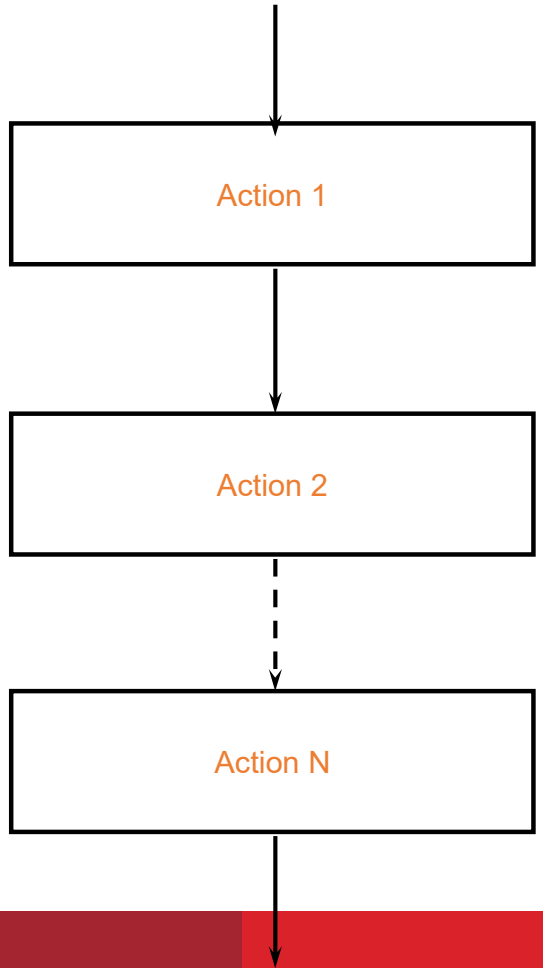


Control Structure: Sequence

Definition: Instructions executed one after another in order.

Code Example:

```
int x=5;
int y=10;
int sum=x+y;
printf("%d", sum);
```



Control Structure: Selection/ Choice

Definition: Executes different actions based on conditions.

Types: if, if-else, nested if, switch/case.

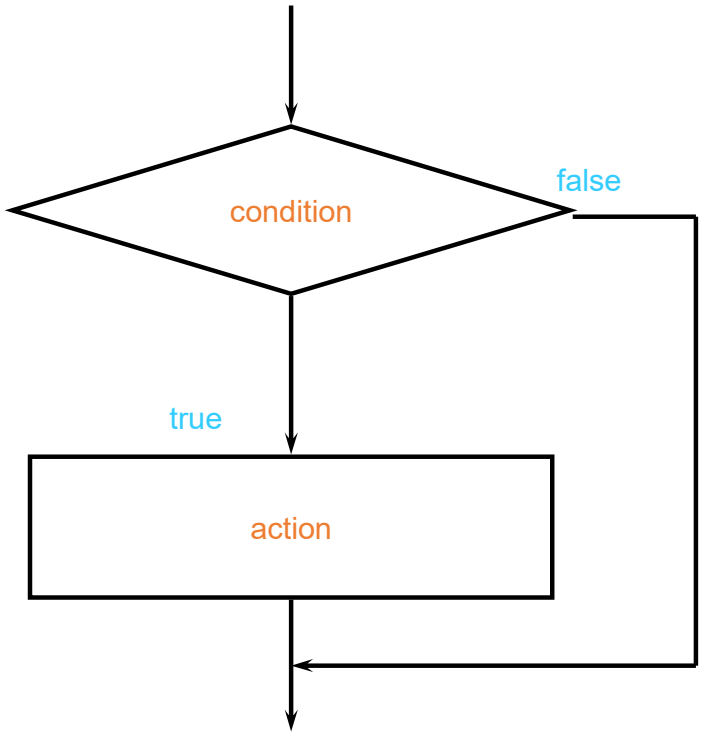
Example: if (grade >= 60) printf("Passed"); else printf("Failed");

Syntax:

```
if(condition)
    action
```

if the condition is true then execute the action.

- action is either a single statement or a group of statements within braces.

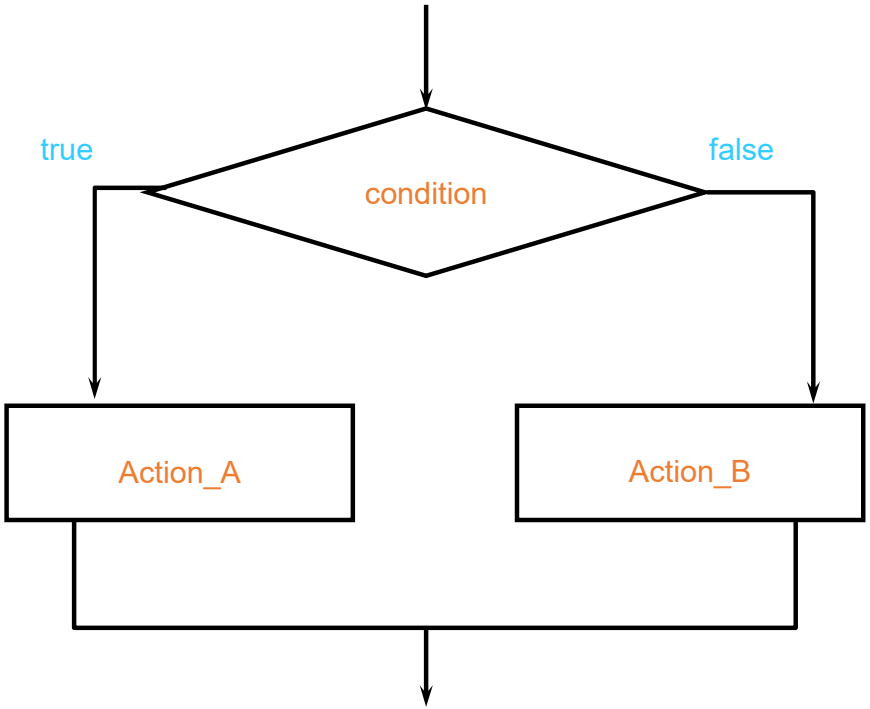


Another Example of Choice Statement

Syntax:

```
if (condition)
    Action_A
else
    Action_B
```

if the condition is true
execute Action_A
else
execute Action_B.



Control Structure: Repetition/Loop

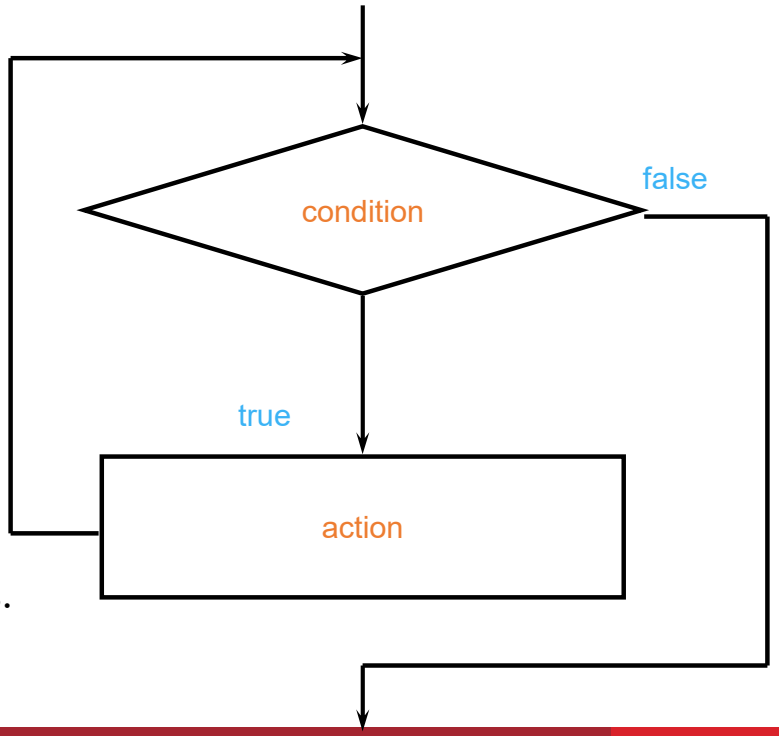
Definition: Repeats a block of code while a condition is true.
Types: while, do-while, for loops.
Example: while (count < 10)

```
 {
 count++;
 printf("%d", count);
}
```

Syntax

```
while (condition)
  action
```

How it works:
if condition is true then execute action
repeat this process until condition evaluates to false action is either a single statement or a group of statements within braces.

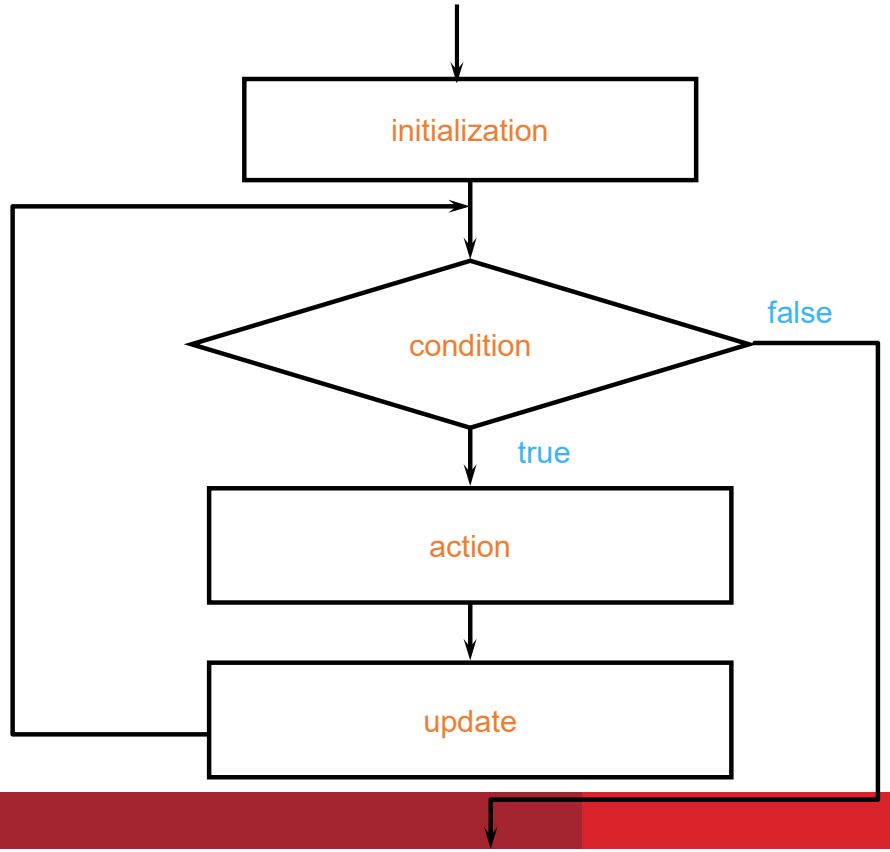


Example of Loop Statement:

Syntax:

```
for (initialization; condition; update)
    action
```

How it works:
execute initialization
  statement
while condition is true
execute action
execute update



Another Example of Loop Statement:

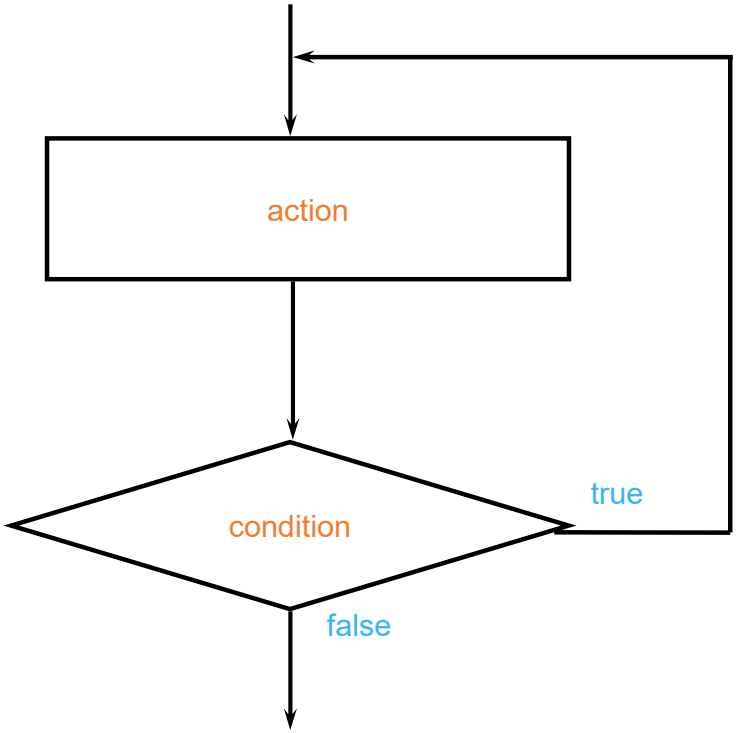
Syntax:

```
do action
while (condition)
```

How it works:

- Execute action if condition is true then execute action again repeat this process until condition evaluates to false.

- action is either a single statement or a group of statements within braces.



Functions in Structured Programming

Introduction:

- We divide the whole program into small Block called functions.
- Function is a block of statements that are executed to perform a specific task.
- It plays an important role in structured programming.

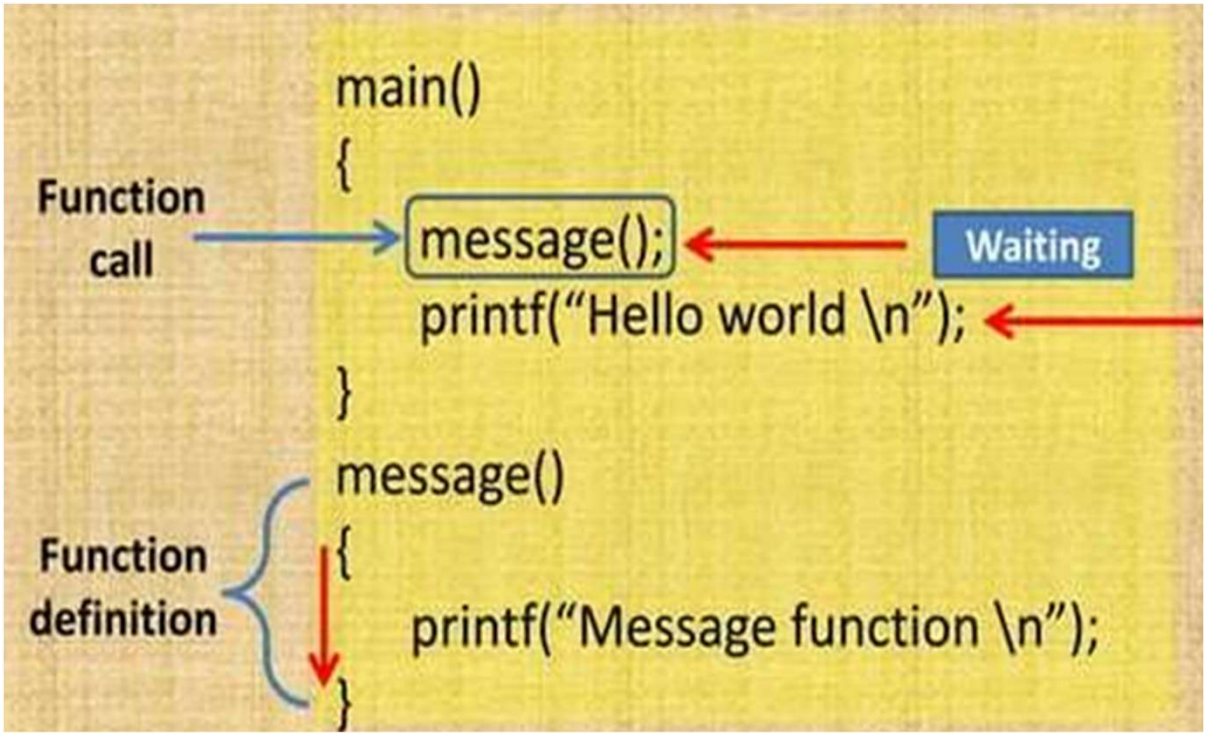
Function call :

- When a function is called by its name, the control moves to the function definition and execute all the statements written in it. Semicolon is used after the name of functions

Function definition

- It contains the instruction that are executed for function is called. Semicolon is not used after the name of the function

Functions Call and Definition



Functions Call

```
main()
{
printf("During Summer ");
wakeup();
milk();
Study();
Play();
Sleep();
system("pause");
}
```

Functions Definition

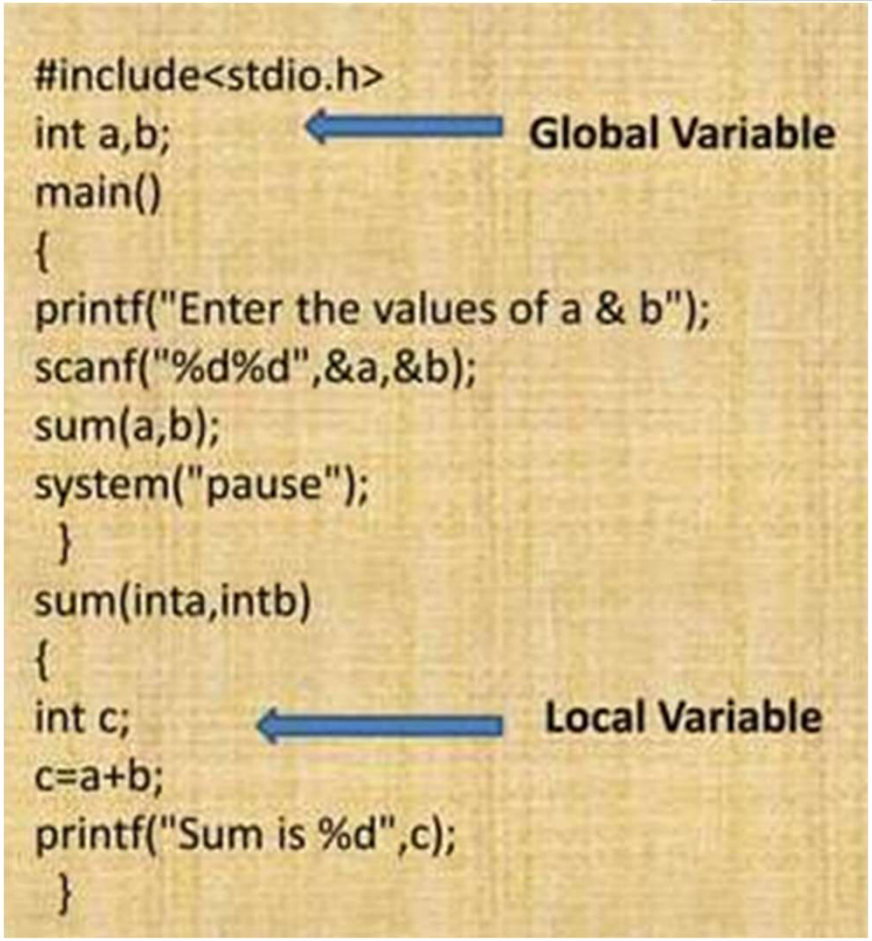
```
study()
{
printf("I study at 2'pm");
}
wakeup()
{
printf("I wake up at 6'am");
}
milk()
{
printf("I drink milk at 8'am");
}
Sleep()
{
printf("I sleep at 9'pm");
}
Play()
{
printf("I play at 3'pm");
}
```

Important Points to remember:

- C program must contains at least one function
- Execution of C programmes begin with main() function.
- If there are more than one function, then one function must be main().
- There is no limit to number of functions in C program.
- Functions in a program are called in sequence as mentioned in main() function.
- After the execution of function, control returns to main()
- The function can call itself. Such a process is called recursion.

Variable Scope

- Variable scope determine the area in a program where variable can be accessed.
- When a variable losses its scope. It means it's data value is lost.
- Common types of variables in C,
  Local variable & Global Variable
- Global variable can be accessed anywhere in the program.
- Local variable can be accessed only in that function in which it is declared.



Thank You