ABES Engineering College, Ghaziabad

Programming for Problem Solving

B.TECH 1ST YEAR : 2023-2024

Dilip Kumar Bharti

Assistant Professor

CSE-DS DEPARTMENT

ABES, Engineering College, Ghaziabad, Uttar Pradesh.

FUNCTION

Functions in C Programming:

Definition:

Functions are blocks of code that perform a specific task.

Purpose:

Modularize code, improve readability, and facilitate code reuse.

Bullet points:

Divide complex tasks into manageable pieces
Reusable code blocks

Function Declaration:

Syntax: return_type function_name(parameters);
Example:

```
int add(int a, int b);
```

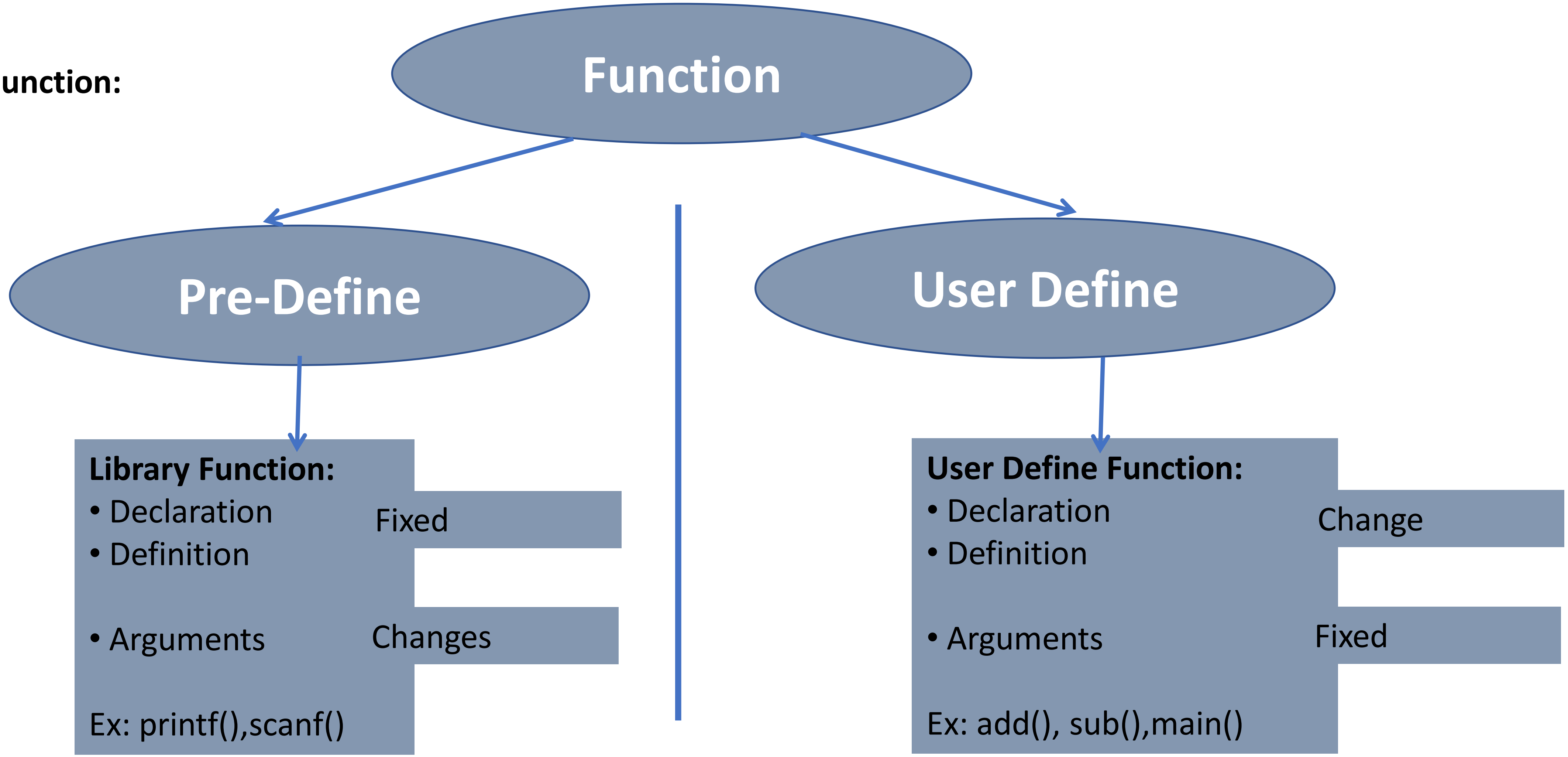
Function Definition
Example:

```
int add(int a, int b)
{
return a + b;
}
```

Calling Functions
Example:

```
result = add(5, 3);
```

Type of Function:



Function

Pre-Define

User Define

Library Function:
- Declaration
- Definition

Fixed

- Arguments

Changes

Ex: printf(),scanf()

User Define Function:
- Declaration
- Definition

Change

- Arguments

Fixed

Ex: add(), sub(),main()

Programming for Problem Solving

ABES Engineering College

Aspects Of Function:

1. Function Declaration.
2. Function Calling
3. Function Definition

Function Declaration :

Compiler Know the

1. Function Name.
2. Number of Parameters
3. Data type of parameters
4. Return type of the function

Function Calling:

- Calling a function to be execute by Compiler
- The only thing to take care of this that you need to pass as many arguments of the same data type as mentioned while declaration the function.

Function Definition:

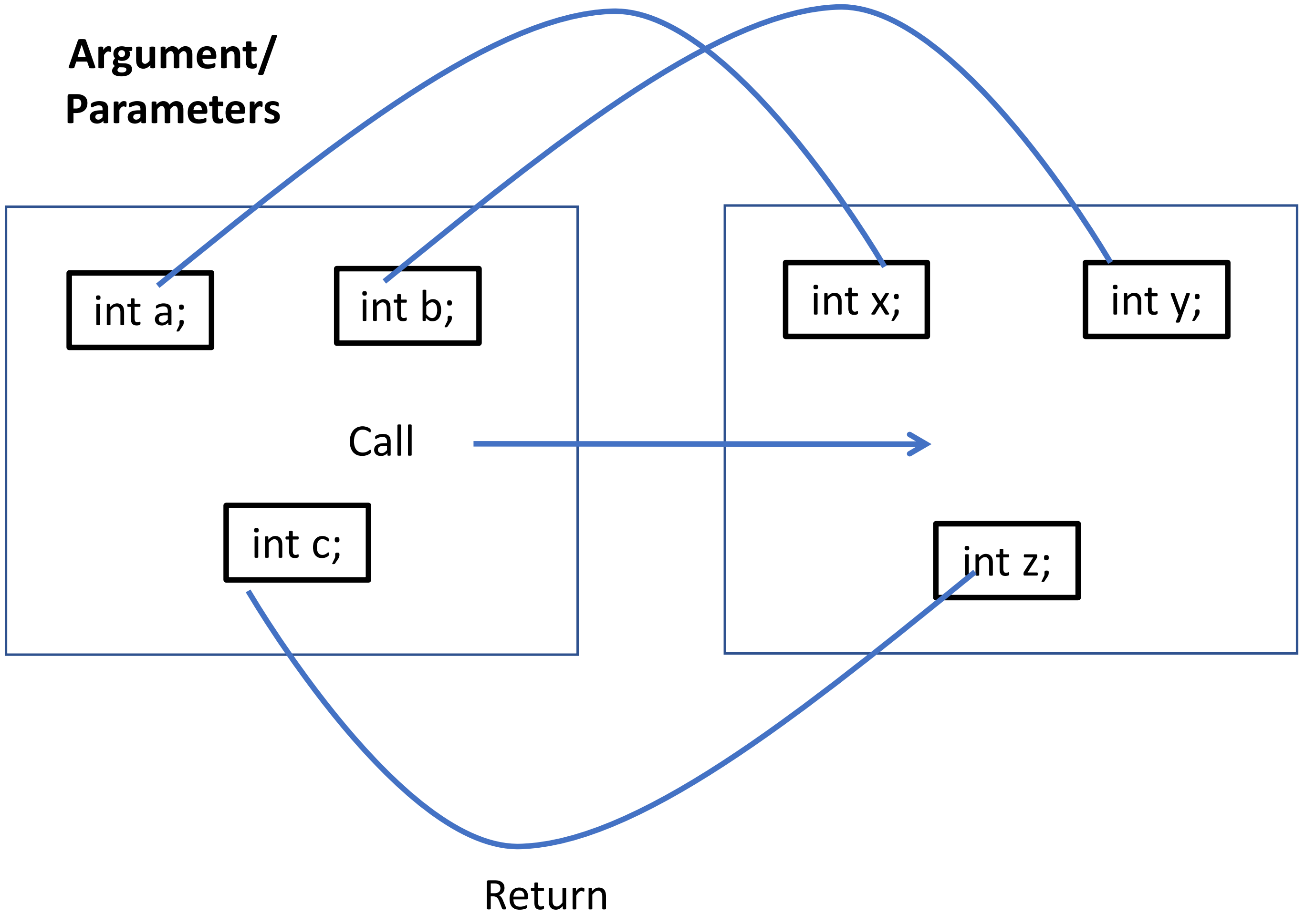
- it is defining the actual statements that the compiler will execute upon calling the function.

Example:

```
#include<stdio.h>

void add(int,int);                          Declaration of Function
int main()
{
        int a,b;
        printf(“Enter Two Number:”)’
        scanf(“%d%d”,&a,&b);
        add(a,b)                            Function Calling
        return 0;
}
void add(int a,int b)
{
        int c;
        c=a+b;                              Definition of Function
        printf(“%d + %d = %d”,a,b,c);
}
```

Programming for Problem Solving



Communication Way of Function:

1. No return with No Argument..
2. No return with Arguments.
3. Return with No Arguments.
4. Return with Arguments.

Arguments Vs Parameters:

Arguments:

- Also known as actual parameters.
- Arguments are used while calling the function.
- Arguments is the Actual value.

Example:

```
int main()
{
		int a;
		a=200;
		call(a);
		return 0;
}
```

Parameters:

- Also known as formal parameters.
- Parameters are used during the declaration of the function.
- Parameters is variable.

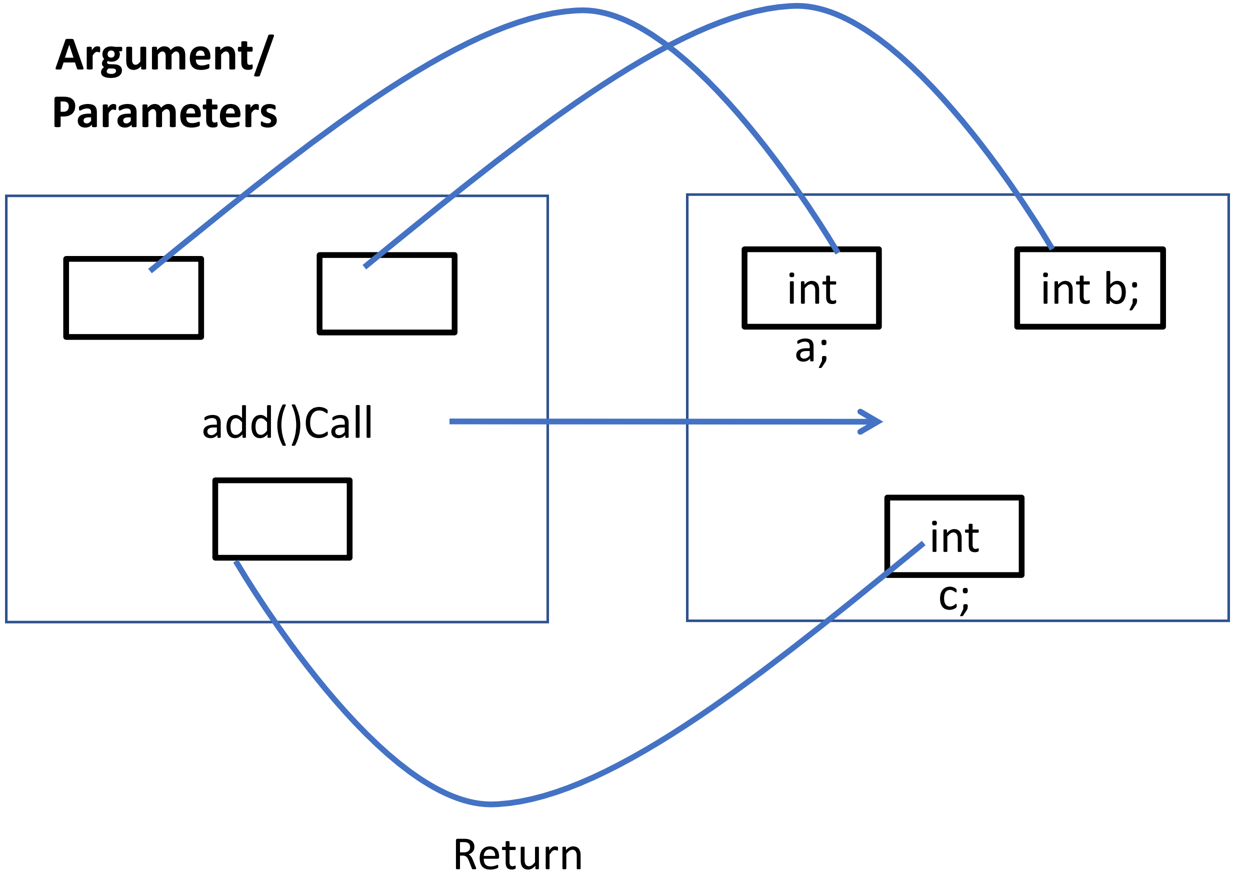
Example:

```
Void add(int,int);
                  or
void add(int a,int b);
```

Programming for Problem Solving

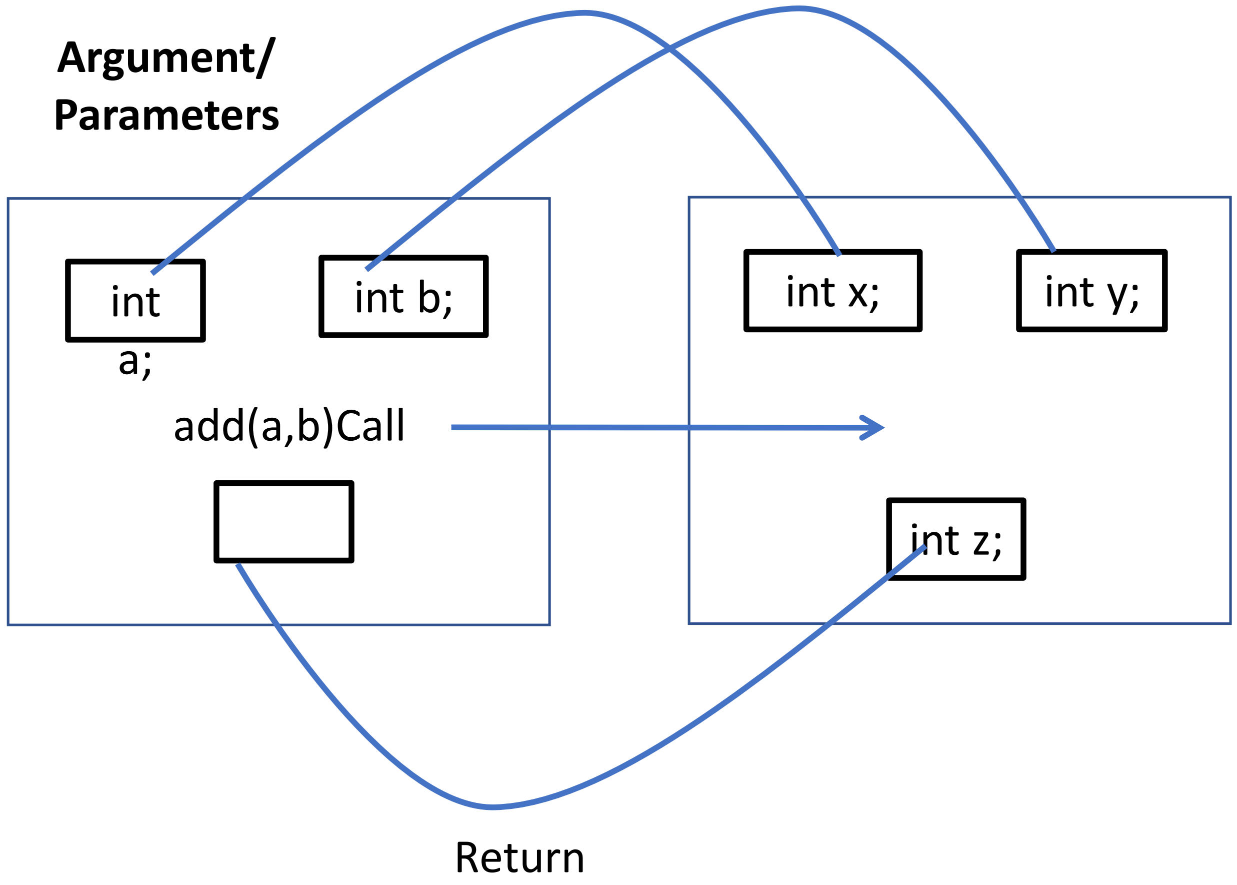
1. **No return with No Argument..**

```
#include<stdio.h>
void add();
int main()
{
      add();
      return 0;
}
void add()
{
      int a,b,c;
      printf("Enter Two Number:");
      scanf("%d%d",&a,&b);
      c=a+b;
      printf("%d + %d = %d",a,b,c);
}
```



2. No return with Argument..

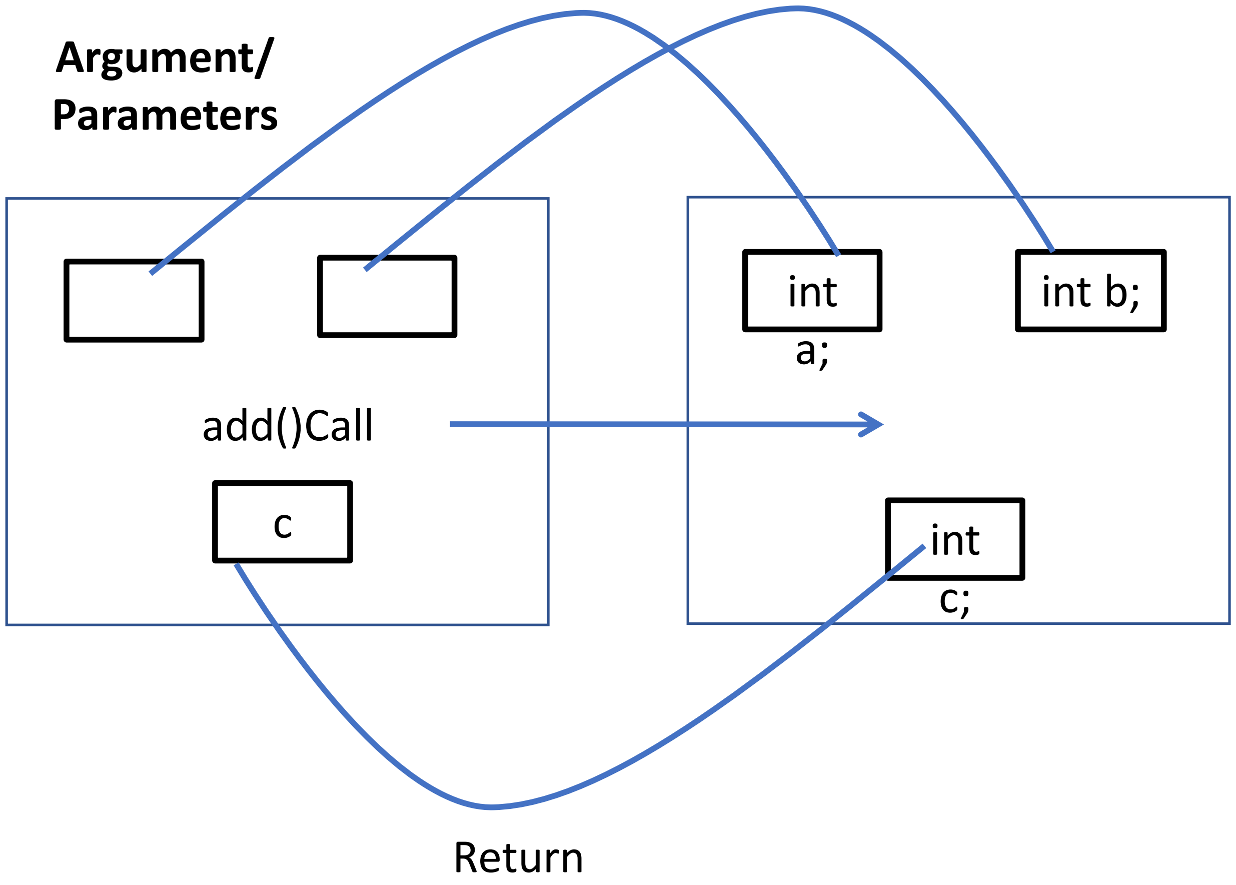
```
#include<stdio.h>
void add(int,int);
int main()
{
      int a,b;
      printf(“Enter Two Number:”);
      scanf(“%d%d”,&a,&b);
      add(a,b);
      return 0;
}
void add(int x, int y)
{
      int z;
      z=x+y;
      printf(“%d + %d = %d”,x,y,z);
}
```



Programming for Problem Solving

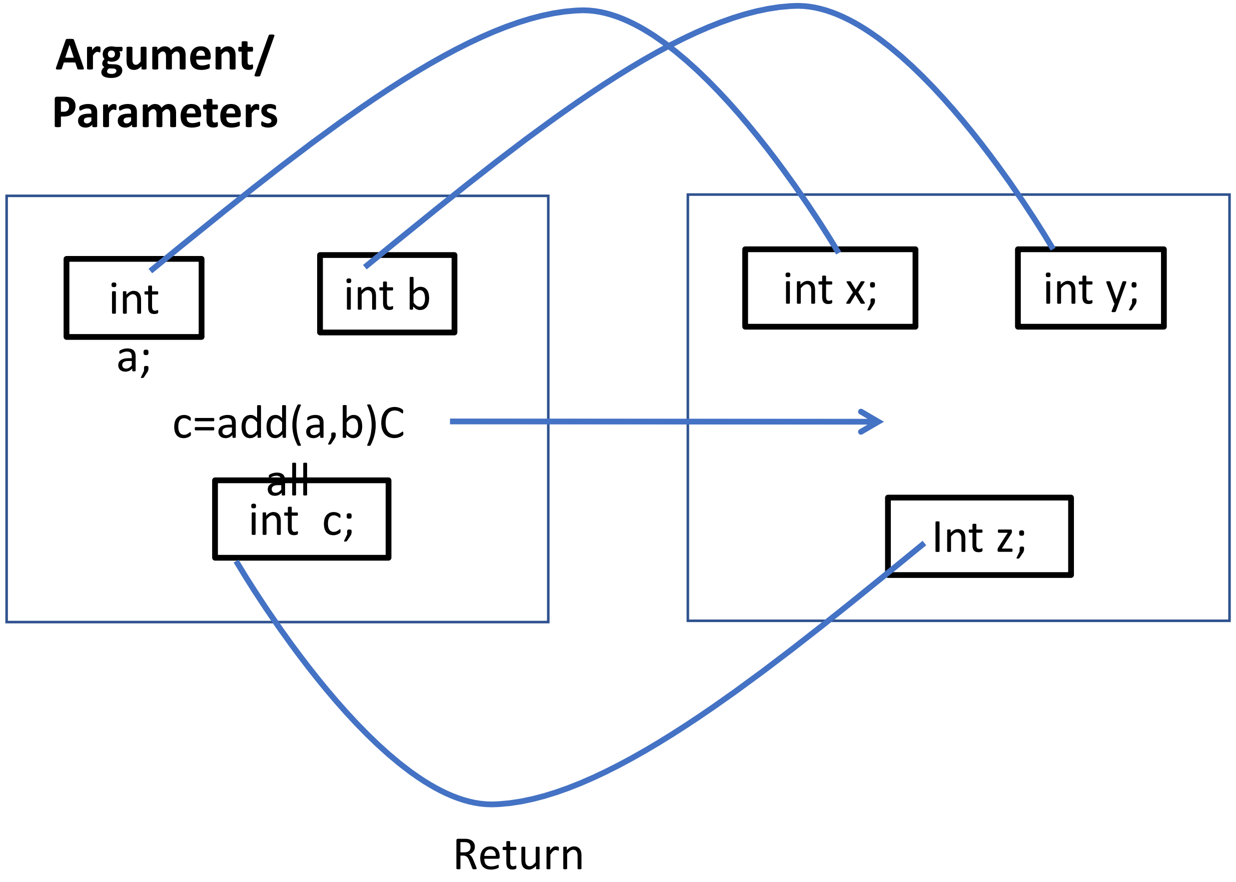
3. Return with No Argument..

```
#include<stdio.h>
int add();
int main()
{
      printf("Sum= %d",add());
      return 0;
}
int add()
{
      int a,b,c;
      printf("Enter Two Number:");
      scanf("%d%d",&a,&b);
      c=a+b;
      return c;
}
```



4. Return with Argument.

```
#include<stdio.h>
int add(int,int);
int main()
{
      int a,b,c;
      printf("Enter Two Number:");
      scanf("%d%d",&a,&b);
      c=add(a,b);
      printf("%d + %d = %d",a,b,c);
      return 0;
}
void add(int x,int y)
{
      int z;
      z=x+y;
      return z;
}
```



Programming for Problem Solving

Practice Questions:

Example: DAY08/01

Problem Statement:

You are given a positive integer n, and you need to implement a function compute_sum to calculate the sum of all positive integers less than or equal to n that are divisible by either 3 or 5.

Input:

A positive integer n (1 <= n).

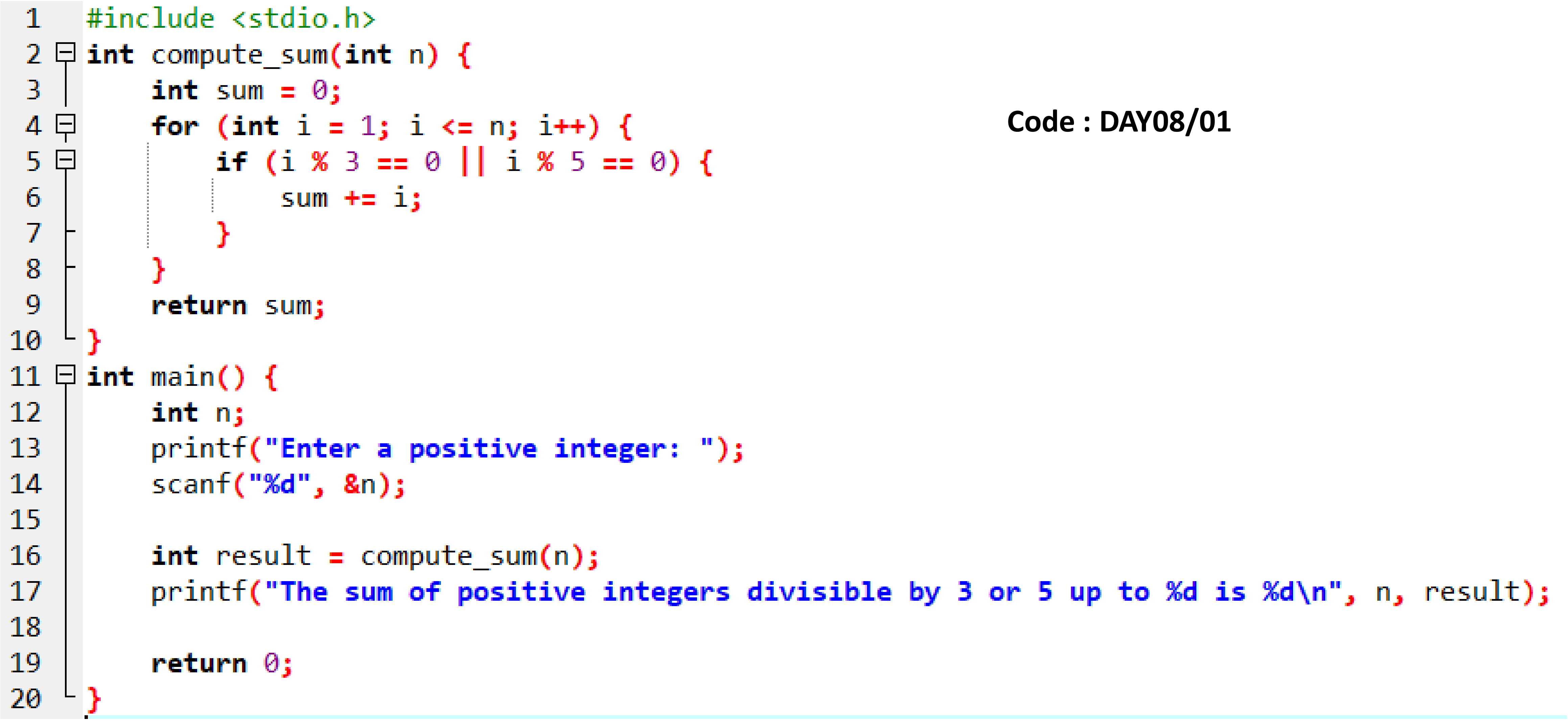
n=10

Output:

An integer, the sum of all positive integers less than or equal to n that are divisible by 3 or 5.

33

Programming for Problem Solving



```c
#include <stdio.h>
int compute_sum(int n) {
    int sum = 0;
    for (int i = 1; i <= n; i++) {
        if (i % 3 == 0 || i % 5 == 0) {
            sum += i;
        }
    }
    return sum;
}
int main() {
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);

    int result = compute_sum(n);
    printf("The sum of positive integers divisible by 3 or 5 up to %d is %d\n", n, result);

    return 0;
}
```

Code : DAY08/01

Programming for Problem Solving

Practice Problems:

1. Write a C function to find the maximum of three integers using conditional statements (if-else).
2. Write a C function to check if a given number is prime or not.
3. Write a C function to check if a given number is even or odd.(Function Signature is return with argument.).
4. Write a C function to check if a given number is palindrome or not.(Function Signature is return with argument.).
5. Write a C function to check if a given number is Armstrong or not.

THANK YOU

* * * * *