

## C-86 $\Rightarrow$ Functions in C $\Rightarrow$ Part 3

### Functions Definitions

```
void sum();  
void main()  
{
```

$\rightarrow$  function declaration

```
    sum();  
}
```

$\rightarrow$  Calling function

```
void sum()  
{
```

$\rightarrow$  called function

```
int a, b, sum = 0;  
printf("Enter a two numbers:");  
scanf("%d %d", &a, &b);  
sum = a + b;  
printf("sum = %d", sum);  
}
```

## Function Declaration :-

Function  
Header

return type    functionname (arguments with datatype)

Function  
Body

```

{
    local variables;           optional
    statements;
    return statements;         optional
}
    
```

\* We can skip local variables and we can write some statements and return it.

\* We can use local variables and can skip return statements.

## Function Declaration :-

return type    functionname (datatype)

Eg ⊕ void sum (int, int);

void main()

{

sum(5, 7);    printf("Hello\n");

}

void sum (int a, int b)

{

int sum = 0;

sum = a + b;

printf("sum = %d", sum);

}



Q2 void sum(int, int); → no need to write name of argument  
 void main()  
 {

int x=5, y=5;  
 sum(x, y); → name of argument or values  
 printf("Hello\n");  
 }

void sum(int a, int b)  
 {

int sum=0;  
 sum = a+b;  
 printf("sum = %d", sum);  
 }

Arguments :- In function calling we say as arguments. (or) actual parameters.

Parameters :- In function declaration & definition we say as parameters (or) formal parameters.

\* { void sum(int, int); → correct } any one  
 { void sum(int a, int b); → correct }

\* { void sum(int a, int b); } same name optional but  
 void sum(int a, int b); } datatype should match.  
 { ---;  
 }

void sum(int a, int b);  
 void sum(int x, int y);  
 {  
 ---;  
 }

void sum(int, float);  
 void sum(int x, float y);

Date \_\_\_\_\_  
Page \_\_\_\_\_

\* Order of datatype should match but it is not necessary to match name of parameters.

\* If we don't specify the return type of function, then compiler will automatically consider it as integer.

Assignment:

Which is function declaration, function definition and function call?

① `int fact(int a);`  
    ↳ function declaration

⑧ `fact(int);`  
    function declaration  
    with error  
    (no return type)

② `void multiply(int, int);`  
    ↳ statement blocks;  
    ↳ function definition

③ `sum(x, y);` function call

④ `divide(m+5, n+5);` function call

⑤ `sum(x+2, 10)` function call with error (no semicolon)

⑥ `void isPrime(int);` function declaration

⑦ `void sum(int a, float b)` function definition  
    ↳  
    `printf("sum = %.f", a+b);`  
    ↳ `return;`