

C-85 ⇒ Functions in C ⇒ Part 2

Function Declaration

* Before using variables we declare those variables.

Example :

```
void sum();  
{  
    int a, b, sum = 0; → variable declaration  
    printf("Enter 2 nos:");  
    scanf("%d %d", &a, &b);  
    sum = a + b;  
    printf("%d", sum);  
}
```

* Similarly before using functions, it is a good practise to declare the functions but it not necessary.

Function Declaration :

return type ^{function name} → no argument
void sum(); → function declaration
void sum(void);
void main()
{

Best use

```
    sum();  
}  
void sum()  
{
```

```
    int a, b, sum = 0;  
    printf("Enter 2 nos:");  
    scanf("%d %d", &a, &b);  
    sum = a + b;  
    printf("sum = %d", sum);  
}
```

Function Declaration:-

return type function name (^{data} type of argument);

void sum (void);

* Always specify type of argument; if we don't pass any argument then in declaration put void;

* If we pass any argument in function calling; but in function declaration we don't specify it ^{this it won't be passed} (or) if we put void then we get error.

Example ①

void sum (void); → ^{function} declaration

void main ()

{

 sum (5, 7); → calling function

 ↑

void sum ()

{

 int a, b, sum = 0;

 printf ("Enter 2 nos: ");

 scanf (" %d %d ", &a, &b);

 sum = a + b;

 printf (" %d ", sum);

}

⇓
Error

Example ②

void sum ();

void main ()

{

 sum (5, 7);

 }

void sum ()

{

 int a, b, sum = 0;

 printf ("Enter 2 nos: ");

 scanf (" %d %d ", &a, &b);

 sum = a + b;

 printf (" %d ", sum);

}

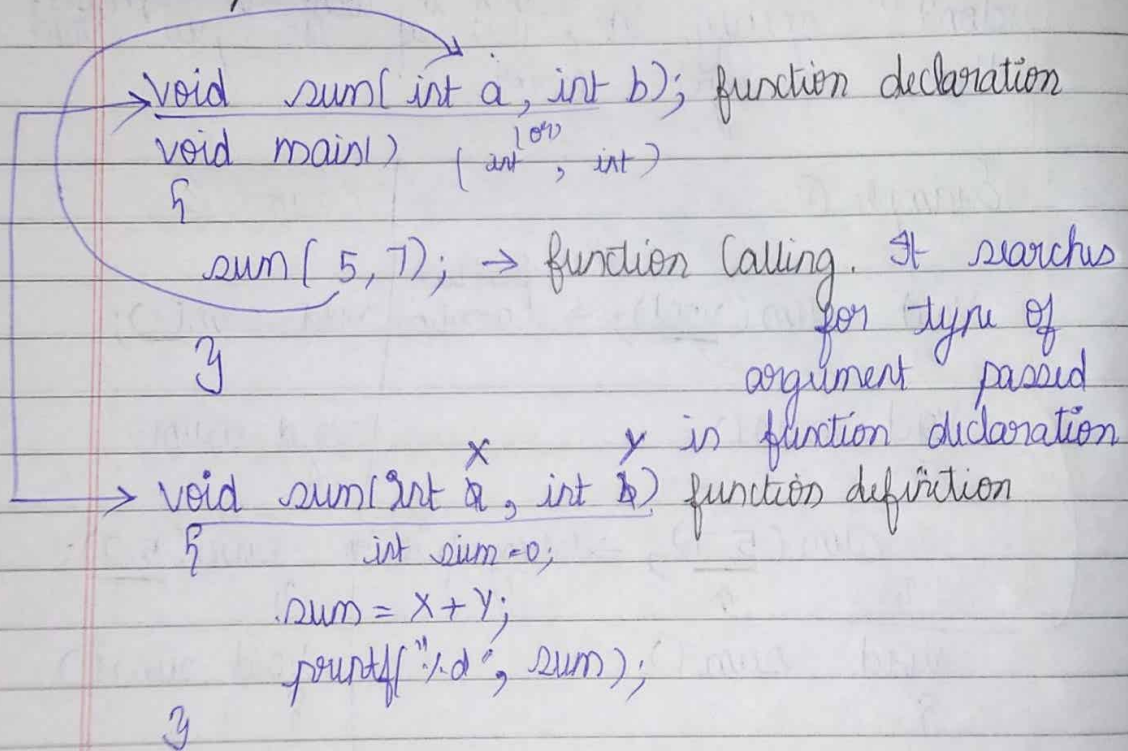
⇓ we get error but (5, 7) not passed; declaration is blank without argument

NOTE:

* In function calling if we don't pass any argument; then in function declaration specify the argument void.

* If in function calling if we pass any argument; then in function declaration specify the type of argument; if we don't specify the argument type or left blank then that arguments will not be passed;

Example (3):-



* Function declaration and definition should match.

It is not always necessary to write the name of argument in function declaration, but always necessary to specify data type in function declaration.

Notes:-

Function declaration will tell compiler about 3 things of function

(i) which type of value the function is going to return after completing execution.
(Return type)

(ii) name of function by using which we call the function.

(iii) Type of arguments and number of arguments we pass in the function.

(iv) In declaration of function; if we don't specify return type; then the compiler by default will assume it as integer type.

Program 1:-

```
void sum(); // function declaration
```

```
void main()
```

```
{
```

```
    sum(); // function calling.
```

```
}
```

```
void sum() // function definition.
```

```
{
```

```
    int a, b, sum = 0;
```

```
    printf("Enter two numbers:");
```

```
    scanf("%d %d", &a, &b);
```

```
    sum = a + b;
```

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

```
}
```


output :-

Enter two numbers:

4

5

sum = 9

Program 2

```
// void sum(); // function declaration  
// put in comment box.
```

```
void main()
```

```
{
```

```
    sum();
```

```
}
```

```
void sum()
```

```
{
```

```
    int a, b, sum = 0;
```

```
    printf("Enter two numbers:");
```

```
    scanf("%d %d", &a, &b);
```

```
    sum = a + b;
```

```
    printf("sum = %d\n", sum);
```

```
}
```

output :-

Enter two numbers:

4

5

sum 9

Even though we get output here, we also get warning message as implicit declaration of function.

Program 3:

```
// void sum();
void main()
{
    sum(); // by default compiler assumes return
           // type for sum() is integer
    float void sum()
    {
```

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

Output: Error (conflicting of function return type)

* Here we don't get o/p because; in function calling compiler assumes it as integer but in function definition return type is float.

* So, here declaration is compulsory.

Program 4:-

```
float sum(void); // function declaration
void main()
{
    sum(); // calling function
    float sum()
    {
        // ...
    }
}
```

O/p:-

Enter 2 nos.:
2.0
2.0
Sum = 4.0.

(Same)

Program 5..

```

float sum(); // function declaration
void main()
{
    sum(5,7); // function calling with argument.
}

float sum()
{
    float a, b, sum = 0;
    printf("Enter 2 numbers:");
    scanf("%f %f", &a, &b);
    sum = a + b;
    printf("sum = %f");
}

```

O/p:

Here argument 5,7 is given in function calling but type of argument is not specified in function declaration; so arguments are not passed and it is ignored and compiler moves to function definition and here we get output without error

```

Enter 2 nos:
6.0
7.0
sum = 13.

```

Program 6: If you don't pass any argument in calling function, then put void datatype in function declaration

```
float sum(void);  
void main()  
{  
    sum(5, 7);  
}
```

error

```
float sum()  
{
```

```
    float a, b, sum = 0;  
    printf("Enter 2 numbers:");  
    scanf("%f %f", &a, &b);    sum = a + b;  
    printf("sum = %f");  
}
```

O/P → Error:

* Here we get error; because the type of arguments ~~must~~ we pass in function calling must be declared in function declaring.

* But here we put type of argument as void instead of Integer in function declaration; hence we get error.

Program 7:-

```
// float sum(void);  
void main()  
{  
    sum();  
}  
int int sum(void)  
{  
    ~~~~~  
}
```

O/P:-

We get o/p but also we get warning for function declaration

Program 8 and 9:-

// char func(); → function declaration

```
void main()
{
    char ch;
    ch = func();
    printf("%c", ch);
}
```

```
char func()
{
    char c;
    printf("Enter a character:");
    scanf("%c", &c);
    return c;
}
```

(9) → No Error
output
Enter character: h
h

Output: Error

* Because; declaration and definition of func() do not match.

* Since we don't declare func(); by default compiler assumes it as integer return type and for definition of func() we have char return type.

* So we get error.

* So in this case; it is must to declare the func() either inside or outside the main function.

Program 10:-

```
char fun();  
void main()  
{
```

```
    char ch;  
    ch = fun();  
    printf("%c", ch);  
}
```

```
char fun()  
{
```

```
    char c;  
    printf("Enter a character");  
    scanf("%c", &c);  
    return 'c';  
}
```

O/P
Enter character: h
c

PROBLEM 1:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 1-FUNCTIONS-PART 2 **/
4  void sum(); //function declaration
5  int main()
6  {
7      sum(); //function calling
8      getch();
9  }
10 void sum() //function definition
11 {
12     int a,b,sum=0;
13     printf("Enter two numbers:\n");
14     scanf("%d %d",&a,&b);
15     sum=a+b;
16     printf("Sum=%d",sum);
17 }
18
```

"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 7_JENNYS LECTUR

Enter two numbers:

2

2

Sum=4

PROBLEM 2:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 2-FUNCTIONS-PART 2 **/
4  // void sum(); //function declaration
5  int main()
6  {
7      sum(); //function calling
8      getch();
9  }
10 void sum() //function definition
11 {
12     int a,b,sum=0;
13     printf("Enter two numbers:\n");
14     scanf("%d %d",&a,&b);
15     sum=a+b;
16     printf("Sum=%d",sum);
17 }
18
```

Logs & others

File	Line	Message
=== Build: Debug in 2_FUNCTIONS_PART 2 (compiler: GNU GCC Compiler...		
D:\1. C NO...		In function 'main':
D:\1. C NO...	7	warning: implicit declaration of function 'sum' [-Wimplicit-functi...
D:\1. C NO...	8	warning: implicit declaration of function 'getch'; did you mean 'g...
D:\1. C NO...	10	warning: conflicting types for 'sum'
D:\1. C NO...	7	note: previous implicit declaration of 'sum' was here

"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 7_JENNYS LECTURE_FUNCTION

```
Enter two numbers:
2
2
Sum=4
```


PROBLEM 3:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 3-FUNCTIONS-PART 2 **/
4  //void sum(); //function declaration
5  int main()
6  {
7      sum(); //function calling
8      getch();
9  }
10 float sum() //function definition
11 {
12     int a,b,sum=0;
13     printf("Enter two numbers:\n");
14     scanf("%f %f",&a,&b);
15     sum=a+b;
16     printf("Sum=%f",sum);
17 }
18
```

Logs & others

File	Line	Message
=== Build: Debug in 3_FUNCTIONS_PART 2 (compiler: GNU GCC Compiler...		
In function 'main':		
D:\1. C NO...	7	warning: implicit declaration of function 'sum' [-Wimplicit-functi...
D:\1. C NO...	8	warning: implicit declaration of function 'getch'; did you mean 'g...
D:\1. C NO...	10	error: conflicting types for 'sum'
D:\1. C NO...	7	note: previous implicit declaration of 'sum' was here

PROBLEM 4:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 4-FUNCTIONS-PART 2 **/
4  float sum(); //function declaration
5  int main()
6  {
7      sum(); //function calling
8      getch();
9  }
10 float sum() //function definition
11 {
12     float a,b,sum=0;
13     printf("Enter two numbers:\n");
14     scanf("%f %f",&a,&b);
15     sum=a+b;
16     printf("Sum=%f",sum);
17 }
18
```

"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 7

Enter two numbers:


4.5

4.5

Sum=9.000000

PROBLEM 5:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 5-FUNCTIONS-PART 2 **/
4  int sum(); //there is no type of argument in function definition
5  int main()
6  {
7      sum(5,7); //we pass argument in function calling
8      getch();
9  }
10 int sum() //since argument type is not defined, its not passed to function defini
11 {
12     int a,b,sum=0;
13     printf("Enter two numbers:\n");
14     scanf("%d %d",&a,&b);
15     sum=a+b;
16     printf("Sum=%d",sum);
17 }
18
```

 "D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 7_JENNYS L

Enter two numbers:

4

5

Sum=9

PROBLEM 6:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 6-FUNCTIONS-PART 2 **/
4  int sum(void); //there is no type of argument in function definition
5  int main()
6  {
7      sum(5,7); //we pass argument in function calling
8      getch();
9  }
10 int sum() //since argument type is not defined, its not passed to function defin
11 {
12     int a,b,sum=0;
13     printf("Enter two numbers:\n");
14     scanf("%d %d",&a,&b);
15     sum=a+b;
16     printf("Sum=%d",sum);
17 }
18
```

Logs & others

File	Line	Message
=== Build: Debug in 6_FUNCTIONS_PART 2 (compiler: GNU GCC Compiler...		
In function 'main':		
D:\1. C NO...	7	error: too many arguments to function 'sum'
D:\1. C NO...	4	note: declared here
D:\1. C NO...	8	warning: implicit declaration of function 'getch'; did you mean 'g...
D:\1. C NO...		In function 'sum':

PROBLEM 7:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 7-FUNCTIONS-PART 2 **/
4  //int sum(void);
5  int main()
6  {
7      sum(); //no argument passed and there is no function definition
8      getch();
9  }
10 int sum(void) //void means null, so do not mention in function definition
11 {
12     int a,b,sum=0;
13     printf("Enter two numbers:\n");
14     scanf("%d %d",&a,&b);
15     sum=a+b;
16     printf("Sum=%d",sum);
17 }
18
```

Logs & others

File	Line	Message
=== Build: Debug in 7_FUNCTIONS_PART 2 (compiler: GNU GCC Compiler...		
D:\1. C NO...		In function 'main':
D:\1. C NO...	7	warning: implicit declaration of function 'sum' [-Wimplicit-functi...
D:\1. C NO...	8	warning: implicit declaration of function 'getch'; did you mean 'g...
D:\1. C NO...		In function 'sum':
D:\1. C NO...	17	warning: control reaches end of non-void function [-Wreturn-type]

"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 7_JENNYS LECTURE_FUNCTION

```
Enter two numbers:
2
2
Sum=4
```


PROBLEM 8:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 8-FUNCTIONS-PART 2 **/
4  void main()
5  {
6  char ch;
7  ch=fun(); //function calling ; there is no function declaration; so error
8  printf("Character is %c\n",ch);
9  getch();
10 }
11 char fun()
12 {
13 char ch;
14 printf("Enter character:\n");
15 scanf("%c",ch);
16 return ch;
17 }
18
```


Logs & others

Code::Blocks x Search results x Cccc x Build log x Build messages x CppCheck/Vera++ x CppCheck/Vera++ x

File	Line	Message
D:\1. C NO...	4	=== Build: Debug in 8_FUNCTIONS_PART 2 (compiler: GNU GCC Compiler...
D:\1. C NO...		warning: return type of 'main' is not 'int' [-Wmain]
D:\1. C NO...	7	In function 'main':
D:\1. C NO...	7	warning: implicit declaration of function 'fun' [-Wimplicit-functi...
D:\1. C NO...	9	warning: implicit declaration of function 'getch'; did you mean 'g...
D:\1. C NO...	11	error: conflicting types for 'fun'

PROBLEM 9:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 9-FUNCTIONS-PART 2 **/
4  char fun();
5  void main()
6  {
7      char ch;
8      ch=fun();
9      printf("Character is %c\n",ch);
10     getch();
11 }
12 char fun()
13 {
14     char ch;
15     printf("Enter character:\n");
16     scanf("%c",&ch);
17     return ch;
18 }
19
```

 "D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lecture


Enter character:

a

Character is a

PROBLEM 10:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 10-FUNCTIONS-PART 2 **/
4  char fun();
5  void main()
6  {
7      char ch;
8      ch=fun();
9      printf("Character is %c\n",ch);
10     getch();
11 }
12 char fun()
13 {
14     char ch;
15     printf("Enter character:\n");
16     scanf("%c",&ch);
17     return 'C';
18 }
19
```

 "D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys

Enter character:

A

Character is C