**Types of User Defined Functions in C**

A user-defined function is one that is defined by the user when writing any program, as we do not have library functions that have predefined definitions. To meet the specific requirements of the user, the user has to develop his or her own functions. Such functions must be defined properly by the user. There is no such kind of requirement to add any particular library to the program.

**Different Types of User-defined Functions in C**

There are four types of user-defined functions divided on the basis of arguments they accept and the value they return:

1. Function with no arguments and no return value
2. Function with no arguments and a return value
3. Function with arguments and no return value
4. Function with arguments and with return value

**1. Function with No Arguments and No Return Value**

Functions that have no arguments and no return values. Such functions can either be used to display information or to perform any task on global variables.

**Syntax**

// void return type with no arguments

void function\_name()

{

// no return value

return;

}

**Example**

* C

|  |
| --- |
| // C program to use function with  // no argument and no return values  #include <stdio.h>    **void** sum()  {  **int** x, y;  **printf**("Enter x and y\n");  **scanf**("%d %d", &x, &y);  **printf**("Sum of %d and %d is: %d", x, y, x + y);  }    // Driver code  **int** main()  {      // function call      sum();    **return** 0;  } |

**Output**

Enter x and y

Sum of 4195522 and 0 is: 4195522

**Explanation**

In the above program, function sum does not take any arguments and has no return values. It takes x and y as inputs from the user and prints them inside the void function.

**2. Function with No Arguments and With Return Value**

Functions that have no arguments but have some return values. Such functions are used to perform specific operations and return their value.

**Syntax**

return\_type function\_name()

{

// program

return value;

}

**Example**

* C

|  |
| --- |
| // C program to use function with  // no argument and with return values  #include <stdio.h>    **int** sum()  {  **int** x, y, s = 0;  **printf**("Enter x and y\n");    **scanf**("%d %d", &x, &y);      s = x + y;  **return** s;  }    // Driver code  **int** main()  {      // function call  **printf**("Sum of x and y is %d", sum());  **return** 0;  } |

**Output**

Enter x and y

Sum of x and y is 4195536

**Explanation**

In the above program, function sum does not take any arguments and has a return value as an integer type. It takes x and y as inputs from the user and returns them.

**3. Function With Arguments and No Return Value**

Functions that have arguments but no return values. Such functions are used to display or perform some operations on given arguments.

**Syntax**

void function\_name(type1 argument1, type2 argument2,...typeN argumentN)

{

// Program

return;

}

**Example**

* C

|  |
| --- |
| // C program to use function with  // argument and no return values  #include <stdio.h>    **void** sum(**int** x, **int** y)  {  **printf**("Sum of %d and %d is: %d", x, y, x + y);  }    // Driver code  **int** main()  {  **int** x, y;  **printf**("Enter x and y\n");    **scanf**("%d %d", &x, &y);        // function call      sum(x, y);    **return** 0;  } |

**Output**

Enter x and y

Sum of 0 and 0 is: 0

**Explanation**

In the above program, function sum does take x and y as arguments and has no return value. The main function takes x and y as inputs from the user and calls the sum function to perform the print operation on the given arguments.

**4. Function With Arguments and With Return Value**

Functions that have arguments and some return value. These functions are used to perform specific operations on the given arguments and return their values to the user.

**Syntax**

return\_type function\_name(type1 argument1, type2 argument2,...typeN argumentN)

{

// program

return value;

}

**Example**

* C

|  |
| --- |
| // C program to use function with  // argument and with return values  #include <stdio.h>    **int** sum(**int** x, **int** y) { **return** x + y; }    // Driver code  **int** main()  {  **int** x, y;  **printf**("Enter x and y\n");  **scanf**("%d %d", &x, &y);        // function call  **printf**("Sum of %d and %d is: %d", x, y, sum(x, y));    **return** 0;  } |

**Output**

Enter x and y

Sum of 0 and 0 is: 0

**Explanation**

In the above program, function sum takes two arguments as x and y, and has a return value as an integer type. The main function takes input x and y from the user and calls the sum function to perform a specific operation on the given arguments and returns the value.