C Input Output (I/O)

**In this tutorial, you will learn to use scanf() function to take input from the user, and printf() function to display output to the user.**

**C Output**

In C programming, printf() is one of the main output function. The function sends formatted output to the screen. For example,

**Example 1: C Output**

1. #include <stdio.h>
2. int main()
3. {
4. // Displays the string inside quotations
5. printf("C Programming");
6. return 0;
7. }

**Output**

C Programming

How does this program work?

* All valid C programs must contain the main() function. The code execution begins from the start of the main() function.
* The printf() is a library function to send formatted output to the screen. The function prints the string inside quotations.
* To use printf() in our program, we need to include stdio.h header file using #inclue <stdio.h> statement.
* The return 0; statement inside the main() function is the "Exit status" of the program. It's optional.

**Example 2: Integer Output**

1. #include <stdio.h>
2. int main()
3. {
4. int testInteger = 5;
5. printf("Number = %d", testInteger);
6. return 0;
7. }

**Output**

Number = 5

We use %d format specifier to print int types. Here, the %d inside the quotations will be replaced by the value of testInteger.

**Example 3: float and double Output**

1. #include <stdio.h>
2. int main()
3. {
4. float number1 = 13.5;
5. double number2 = 12.4;
6. printf("number1 = %f\n", number1);
7. printf("number2 = %lf", number2);
8. return 0;
9. }

**Output**

number1 = 13.500000

number2 = 12.400000

To print float, we use %f format specifier. Similarly, we use %lf to print double values.

**Example 4: Print Characters**

1. #include <stdio.h>
2. int main()
3. {
4. char chr = 'a';
5. printf("character = %c.", chr);
6. return 0;
7. }

**Output**

character = a

To print char, we use %c format specifier.

**C Input**

In C programming, scanf() is one of the commonly used function to take input from the user. The scanf() function reads formatted input from the standard input such as keyboards.

**Example 5: Integer Input/Output**

1. #include <stdio.h>
2. int main()
3. {
4. int testInteger;
5. printf("Enter an integer: ");
6. scanf("%d", &testInteger);
7. printf("Number = %d",testInteger);
8. return 0;
9. }

**Output**

Enter an integer: 4

Number = 4

Here, we have used %d format specifier inside the scanf() function to take int input from the user. When the user enters an integer, it is stored in the testInteger variable.

Notice, that we have used &testInteger inside scanf(). It is because &testInteger gets the address of testInteger, and the value entered by the user is stored in that address.

**Example 6: Float and Double Input/Output**

1. #include <stdio.h>
2. int main()
3. {
4. float num1;
5. double num2;
6. printf("Enter a number: ");
7. scanf("%f", &num1);
8. printf("Enter another number: ");
9. scanf("%lf", &num2);
10. printf("num1 = %f\n", num1);
11. printf("num2 = %lf", num2);
12. return 0;
13. }

**Output**

Enter a number: 12.523

Enter another number: 10.2

num1 = 12.523000

num2 = 10.200000

We use %f and %lf format specifier for float and double respectively.

**Example 7: C Character I/O**

1. #include <stdio.h>
2. int main()
3. {
4. char chr;
5. printf("Enter a character: ");
6. scanf("%c",&chr);
7. printf("You entered %c.", chr);
8. return 0;
9. }

**Output**

Enter a character: g

You entered g.

When a character is entered by the user in the above program, the character itself is not stored. Instead, an integer value (ASCII value) is stored.

And when we display that value using %c text format, the entered character is displayed. If we use %d to display the character, it's ASCII value is printed.

**Example 8: ASCII Value**

1. #include <stdio.h>
2. int main()
3. {
4. char chr;
5. printf("Enter a character: ");
6. scanf("%c", &chr);
7. // When %c is used, a character is displayed
8. printf("You entered %c.\n",chr);
9. // When %d is used, ASCII value is displayed
10. printf("ASCII value is % d.", chr);
11. return 0;
12. }

Output

Enter a character: g

You entered g.

ASCII value is 103.

**I/O Multiple Values**

Here's how you can take multiple inputs from the user and display them.

1. #include <stdio.h>
2. int main()
3. {
4. int a;
5. float b;
6. printf("Enter integer and then a float: ");
8. // Taking multiple inputs
9. scanf("%d%f", &a, &b);
10. printf("You entered %d and %f", a, b);
11. return 0;
12. }

**Output**

Enter integer and then a float: -3

3.4

You entered -3 and 3.400000

**Format Specifiers for I/O**

As you can see from the above examples, we use

* %d for int
* %f for float
* %lf for double
* %c for char

Here's a list of commonly used C data types and their format specifiers.

| Data Type | Format Specifier |
| --- | --- |
| int | %d |
| char | %c |
| float | %f |
| double | %lf |
| short int | %hd |
| unsigned int | %u |
| long int | %li |
| long long int | %lli |
| unsigned long int | %lu |
| unsigned long long int | %llu |
| signed char | %c |
| unsigned char | %c |
| long double | %Lf |