Input and Output statements in c

C Input and Output - printf()/scanf(), and more.

- **Input** means to provide the program with some data to be used in it.
- **Output** means to display data on the screen or write the data to a printer or a file.
- The C programming provides standard library functions to read any given input and display output on the console.

While dealing with input-output operations in C, we use the following two streams:

- **Standard Input (stdin)**
- **Standard Output (stdout)**
- **Standard input** or **stdin** is used for taking input.
- **Standard output** or **stdout** is used for giving output.
- The functions used for standard input and output are present in the **stdio.h** header file.
- Hence, to use those functions, we need to include the **stdio.h** header file in our program, as shown below.

```
#include <stdio.h>
```

Functions Used for Input and Output

C language offers us several built-in functions for performing input/output operations. The following are the functions used for standard input and output:

1. printf() function - **Show Output**
2. scanf() function - **Take Input**
3. getchar() and putchar() function

4. gets() and puts() function

1. The printf() function

- The printf() function is the most used function in the C language.
- This function is defined in the **stdio.h** header file and is used to show output on the console (standard output).

printf() Code Examples

Let's start with a simple example.

1. Print a sentence

Let's print a simple sentence using the printf() function.

```
#include <stdio.h>

int main() {

   // using printf()

   printf("Welcome to Studytonight");

   return 0;

}
```

2. Print an Integer value

We can use the printf() function to print an integer value coming from a variable using the %d **format specifier**.

For example,

```
#include <stdio.h>
int main() {
   int x = 10;
   // using printf()
   printf("Value of x is: %d", x);
   return 0;
}
```

3. Print a Character value

The %c format specifier is used to print character variable values using the printf() function.

```
#include <stdio.h>
int main() {
   // using printf()
   char gender = 'M';
   printf("John's Gender is: %c", gender);
   return 0;
}
```

4. Print a Float and a Double value

In the code example below, we have used the printf() function to print values of a float and double type variable.

For float value we use the %f format specifier and
for double value we use the %lf format specifier.

```
#include <stdio.h>
int main() {
   // using printf()
   float num1 = 15.50;
   double num2 = 15556522.0978678;

   printf("Value of num1 is: %f \n", num1);
   printf("Value of num2 is: %lf", num2);
   return 0;
}
```

The scanf() function

When we want to take input from the user, we use the scanf() function and store the input value into a variable.

Here is the **syntax** for scanf():

scanf("%x", &variable);

where, %x is the format specifier.

- Using the format specifier, we tell the compiler what type of data to expect from the user.
- The & is the **address operator** which tells the compiler the address of the variable so that the compiler can store the user input value at that address.

scanf() Code Examples

Let's start with a simple example.

1. Input Integer value

If we have to take an integer value input from the user, we have to define an integer variable and then use the scanf() function.

```
#include <stdio.h>
int main() {
  // using scanf()
  int user_input;

  printf("Please enter a number: ");
  scanf("%d", &user_input);
  printf("You entered: %d", user_input);

  return 0;
}
```

2. Input Float value

Just like integer value, we can take input for any different datatype. Let's see an example of float type value.

```
#include <stdio.h>
int main() {
  // using scanf()
  float user_input;

  printf("Please enter a decimal number: ");
  scanf("%f", &user_input);
```

```
    printf("You entered: %f", user_input);

    return 0;
}
```

3. Input Character value

Let's see how we can take a simple character input from the user.

```
#include <stdio.h>
int main() {
    // using scanf()
    char gender;

    printf("Please enter your gender (M, F or O): ");
    scanf("%c", &gender);
    printf("Your gender: %c", gender);

    return 0;
}
```