**Introducing C**

**C** is a general-purpose programming language that has been around for nearly 50 years.  
**C** has been used to write everything from operating systems (including Windows and many others) to complex programs like the Python interpreter, Git, Oracle database, and more.  
The versatility of C is by design. It is a low-level language that relates closely to the way machines work while still being easy to learn.  
Understanding how computer memory works is an important aspect of the C programming language.

**Hello World!**

As when learning any new language, the place to start is with the classic "Hello World!" program:

#include <stdio.h>  
  
int main() {  
printf("Hello, World!\n");  
return 0;  
}

Let's break down the code to understand each line:  
**#include <stdio.h>** The function used for generating output is defined in **stdio.h**. In order to use the **printf**function, we need to first include the required file, also called a **header file**.  
  
**int main()** The **main**() function is the entry point to a program. Curly brackets { } indicate the beginning and end of a function (also called a code block). The statements inside the brackets determine what the function does when executed.  
Tap **Try It Yourself** to play around with the code.

**Hello World!**

The **printf**function is used to generate output:

#include <stdio.h>  
  
int main() {  
**printf("Hello, World!\n");**  
return 0;  
}

Here, we pass the text "Hello World!" to it.  
The \n **escape sequence** outputs a newline character. Escape sequences always begin with a backslash \.  
The **semicolon** ; indicates the end of the statement. Each statement must end with a semicolon.  
  
**return 0;** This statement terminates the **main()** function and returns the value 0 to the calling process. The number 0 generally means that our program has successfully executed. Any other number indicates that the program has failed.  
Tap **Try It Yourself** to play around with the code.

**Introducing C**

Coding Blog by : Shyed Shahriar Housaini

1st All in One Coding Website in Bangla Language / বাংলা ভাষা

C is a general-purpose programming language that has been around for nearly 50 years. C has been used to write everything from operating systems (including Windows and many others) to complex programs like the Python interpreter, Git, Oracle database, and more. The versatility of C is by design. It is a low-level language that relates closely to the way machines work while still being easy to learn. Understanding how computer memory works is an important aspect of the C programming language.

সি একটি সাধারণ-উদ্দেশ্যমূলক প্রোগ্রামিং ভাষা যা প্রায় 50 বছর ধরে চলছে। অপারেটিং সিস্টেম থেকে আরও অনেক কিছু লিখতে সি ব্যবহার করা হয়েছে, উদাহরণস্বরূপ (উইন্ডোজ, লিনাক্স, ম্যাক ওএস, আই ওএস এবং আরও অনেকগুলি সহ) এবং পাইথন ইন্টারপ্রেটার, গিট, ওরাকল ডাটাবেস, এবং আরও কিছু জটিল প্রোগ্রাম সি প্রোগ্রামিং ভাষায় লেখা হচ্ছে সি এর বহুমুখিতা নকশা দ্বারা । এটি একটি নিম্ন-স্তরের ভাষা যা মেশিনগুলির কাজ করার পদ্ধতিটির সাথে ঘনিষ্ঠভাবে সম্পর্কিত, যদিও শেখা এখনও সহজ। কম্পিউটার মেমোরি কীভাবে কাজ করে তা বোঝা সি প্রোগ্রামিং ভাষার একটি গুরুত্বপূর্ণ বিষয় ।

Hello World!  
**Hello World!  
As when learning any new language, the place to start is with the classic "Hello World!" program:**

#include <stdio.h>

int main() {

printf("Hello, World!\n");

return 0;

}

Let's break down the code to understand each line:  
  
**#include <stdio.h>** The function used for generating output is defined in stdio.h. In order to use the printf function, we need to first include the required file, also called a - **header file.** -- **int main() or int main(void),** The main() function is the entry point to a program, function statement is an order or action or a verb, functions should have (); after them. Curly brackets { } indicate the beginning and end of a function (also called a code block). The statements inside the brackets determine what the function does when executed.

Hello World!   
The **printf();** function is used to generate output:

#include <stdio.h>

int main(void)

//This is a comment in C - main function, (void) because it does not take in any value.

{

printf("Hello, World!\n");

return 0;

}

Here, we pass the text **"Hello World!"** to it. Hello World!   
The printf function is used to generate output:  
The **\n** ; \n escape sequence outputs a newline character. Escape sequences always begin with a backslash \.  
The semicolon ; indicates the end of the statement. Each statement must end with a semicolon.  
**return 0;** This statement terminates the main() function and returns the value 0 to the calling process.  
The number 0 generally means that our program has successfully executed. Any other number indicates   
that the program has failed.

#include <stdio.h>

#include <stdio.h>

[« Previous page 1](file:///G:\GitCloneFiles\index.html)

**Data Types**

C supports the following basic data types:  
**int:** integer, a whole number.  
**float:** floating point, a number with a fractional part.  
**double:** double-precision floating point value.  
**char:** single character.  
  
The amount of storage required for each of these types varies by platform.  
C has a built-in **sizeof** operator that gives the memory requirements for a particular data type.  
**For example**:

#include <stdio.h>  
  
int main() {  
printf("int: %ld \n", **sizeof(int)**);  
printf("float: %ld \n", **sizeof(float)**);  
printf("double: %ld \n", **sizeof(double)**);  
printf("char: %ld \n", **sizeof(char)**);  
  
return 0;  
}

The program output displays the corresponding size in bytes for each data type.  
The **printf** statements in this program have two **arguments**. The first is the output string with a **format specifier** (%ld), while the next argument returns the **sizeof**value. In the final output, the **%ld** (for long decimal) is replaced by the value in the second argument.  
Note that C does not have a boolean type.  
  
A **printf** statement can have multiple format specifiers with corresponding arguments to replace the specifiers. Format specifiers are also referred to as conversion specifiers.  
We will learn more about format specifiers in the upcoming lessons.