

# Introduction to C



**Instructor:** Maninder Kaur Tatla

**Email:** [maninder.kaur2@sheridancollege.ca](mailto:maninder.kaur2@sheridancollege.ca)

**Course:** PROG20799

# Introduction

- C is a general purpose high level language that was originally developed by **Dennis M. Ritchie** to develop the Unix operating system at Bell Labs.
- C was originally first implemented on the DEC PDP-11 computer in 1972.
- The UNIX operating system, and essentially all UNIX applications programs have been written in C.

# Introduction

- The C has now become a widely used professional language for various reasons:
  - Easy to learn.
  - Structured language.
  - It produces efficient programs.
  - It can handle low-level activities.
  - It can be compiled on a variety of computer platforms.

# Facts About C

- C was invented to write an operating system called UNIX.
- C is a successor of B language which was introduced around 1970.
- The language was formalized in 1988 by the American National Standard Institute (ANSI).
- By 1973, UNIX OS was almost totally written in C.
- Today C is the most widely used System Programming Language.
- Most of the state of the art software have been implemented using C.

# Why to use C?

- C was initially used for system development work, in particular the programs that make-up the operating system.
- C was adopted as a system development language because it produces code that runs nearly as fast as code written in assembly language.
- Some examples of the use of C are:
  - Operating Systems.
  - Language Compilers.
  - Assemblers.
  - Text Editors.
  - Print Spoolers.
  - Network Drivers.
  - Modern Programs.
  - Data Bases.
  - Language Interpreters.

# C - Environment Setup

- Before you start doing programming using C language, you need following two software's available on your computer:
  - Text Editor.
  - The C Compiler.

# Text Editor

- This will be used to type your program.
- Examples of few editors include Windows Notepad, OS Edit command, Epsilon, EMACS, and vim or vi.
- Name and version of text editor can vary on different operating systems.
  - For example Notepad will be used on Windows and vim or vi can be used on windows as well as Linux, or Unix.
- The files you create with your editor are called **source files** and contain program source code.
- The source files for C programs are typically named with the extension **.c**.

# C Compilers

- The source code written in source file is the human readable source for your program.
- It needs to be *compiled*, to turn into machine language so that your CPU can actually execute the program as per instructions given.
- The C programming language compiler will be used to compile your source code into final executable program.
- Most frequently used and free available compiler is *GNU C/C++* compiler.
- We will use **Orwell Dev C++** IDE for our course.
  - It has built-in text editor and C compiler.
  - It is compatible with Win 7 and Win 8.
  - Download it from the following URL:  
<http://sourceforge.net/projects/orwelldvcpp/>



# C Hello World Example

- Let us look at a simple code that would print the words **Hello World!**:

```
#include <stdio.h>

main()
{
    /* my first program in C */
    printf("Hello, World!");
}
```

# C Hello World Example

```
#include <stdio.h>

main()
{
    /* my first program in C */
    printf("Hello, World!");
}
```

- Let us look various parts of the program:
  - The first line of the program `#include <stdio.h>` is a preprocessor command which tells a C compiler to include `stdio.h` file before going to actual compilation.
  - The next line `main()` is the main function where program execution begins.
  - The next line `/*...*/` will be ignored by the compiler and it has been put to add additional comments in the program. So such lines are called comments in the program.
  - The next line `printf(...)` is another function available in C which causes the message `Hello, World!` to be displayed on the screen.

# Tokens in C

- A C program consists of various tokens and a token is either a keyword, an identifier, a constant, a string literal, or a symbol.
- For example, the following C statement consists of five tokens:

```
printf("Hello, World!");
```

- The individual tokens are:

```
printf  
(  
"Hello, World!"  
)  
;
```

# Semicolons (;)

- In C program, the semicolon is a statement terminator.
- That is, each individual statement must be ended with a semicolon.
- For example:

```
printf("Hello, World!");
```

# Comments

- Comments are like helping text in your C program and they are ignored by the compiler.
- They start with `/*` and terminates with the characters `*/` as shown below:

```
/* my first program in C */
```

# Identifiers

- A C identifier is a name **used to identify a variable, function, or any other user-defined item.**
- An identifier starts with a letter **A** to **Z** or **a** to **z** or an underscore **\_** followed by zero or more letters, underscores, and digits (**0** to **9**).
- C does not allow punctuation characters such as **@**, **\$**, and **%** within identifiers.
- C is a case sensitive programming language.
  - Thus, **TotalSalary** and **totalsalary** are two different identifiers in C.

# Keywords

- The following list shows the reserved words in C.
- These reserved words may not be used as constant or variable or any other identifier names.

auto	else	long	switch
break	enum	register	typedef
case	extern	return	union
char	float	short	unsigned
const	for	signed	void
continue	goto	sizeof	volatile
default	if	static	while
do	int	struct	_Packed
double			

# Data Types

- All C compilers support a variety of data types. This enables the programmer to select the appropriate data type as per the need of the application.
- In C programming language, data types refers to a system used for declaring *variables* or *functions*.
- Generally, data is represented using numbers or characters. The numbers may be integers or real.
- The *type* of a variable determines how much space it occupies in storage and how the bit pattern stored is interpreted.



# Data Types

- Data types in C can be classified as follows:

S.N o.	Types and Description
1	<b>Basic Types:</b> They are arithmetic types and consists of the two types: (a) integer types and (b) floating-point types.
2	<b>The type void:</b> The type specifier <code>void</code> indicates that no value is available.
3	<b>Derived Types:</b> They include (a) Pointer types, (b) Array types, (c) Structure types, (d) Union types and (e) Function types.

# Entire Data Types in C

Data Type	Size (bytes)	Range	Format String
char	1	−128 to 127	%c
unsigned char	1	0 to 255	%c
short or int	2 or 4	−32,768 to 32,767	%i or %d
unsigned int	2 or 4	0 to 65535	%u
long	4	−2147483648 to 2147483647	%ld
unsigned long	4	0 to 4294967295	%lu
float	4	3.4 e−38 to 3.4 e + 38	%f or %g
double	8	1.7 e-308 to 1.7 e + 308	%lf
long double	10	3.4 e−4932 to 1.1 e + 4932	%lf

# Integer Types - Example

- To get the exact size of a type or a variable on a particular platform, you can use the `sizeof` operator.
- The expressions `sizeof(type)` yields the storage size of the object or type in bytes.
- Following is an example to get the size of int type on any machine:

# Integer Types - Example

```
#include <stdio.h>

main()
{
    printf("Storage size for int: %d", sizeof(int));
}
```

- Check the output for yourself.

# Floating-Point Types - Example

```
#include <stdio.h>

main()
{
    printf("Storage size for float : %d", sizeof(float));
}
```

- Check the output for yourself.

# C - Variables

- A variable is nothing but a name given to a storage area that our programs can work on.
- Each variable in C has a specific type, which determines the size and layout of the variable's memory:
  - the range of values that can be stored within that memory; and
  - the set of operations that can be applied to the variable.
- The name of a variable can be composed of letters, digits, and the underscore character.
  - It must begin with either a letter or an underscore.
- Upper and lowercase letters are distinct because C is case-sensitive.

# Variable Declaration in C

- All variables must be declared before we use them in C program.
- A declaration specifies a **type**, and contains a list of one or more **variables** of that type as follows:

```
type variable_list;
```

- Here, type must be a valid C data type
- **variable\_list** may consist of one or more identifier names separated by commas.
- Some valid variable declarations along with their definition are shown here:

```
int i, j, k;  
char c, ch;  
float f, salary;  
double d;
```

# Variable Initialization in C

- Variables are initialized (assigned an value) with an equal sign followed by a constant expression.
- The general form of initialization is:

```
variable_name = value;
```

- Variables can be initialized (assigned an initial value) in their declaration.
- The initializer consists of an equal sign followed by a constant expression as follows:

```
type variable_name = value;
```



# Variable Initialization in C

- Some examples are:

```
int d = 3, f = 5;    /* initializing d and f. */  
byte z = 22;        /* initializes z. */  
double pi = 3.14159; /* declares an approximation of pi. */  
char x = 'x';       /* the variable x has the value 'x'. */
```

- It is a good programming practice to initialize variables properly otherwise, sometime program would produce unexpected result.

# Variable Initialization in C

```
#include <stdio.h>

main ()
{
    /* variable declaration: */
    int a, b;
    int c;
    float f;

    /* actual initialization */
    a = 10;
    b = 20;

    c = a + b;
    printf("Value of c : %d \n", c);

    f = 70.0/3.0;
    printf("Value of f : %f \n", f);
}
```

## ■ Output:

```
value of c : 30
value of f : 23.333334
```

# Practice Problems

- Let's try some programs in C language:
  - Program to find sum and average of three numbers.
  - Program to find area and circumference of a circle.
  - Program to find whether a number is even or odd.
  - Program to find largest of three numbers.
  - Program to print a table of any number.
  - Program to print first N prime numbers.
  - Program to find factorial of a number.



**Any questions please?**