

# Overview of C

## INTRODUCTION TO C LANGUAGE

C is a general-purpose programming language that is closely related to how computer machines work. Although often considered difficult to learn, C is actually a simple language with vast capabilities.

Here are some key points to note in C:

- **Case-sensitive:** C distinguishes between uppercase and lowercase letters. For example, `printf` and `Printf` are two different things.
- **Space-insensitive:** Separators such as spaces, tabs, or new lines do not affect the program.
- **Semicolon:** Every statement must end with a semicolon (`;`).
- **Multiple Statements:** Several statements can be written on the same line.

---

## SIMPLE C PROGRAM: PRINTING A LINE OF TEXT

The simplest C program is a program that prints text. Here is an example:

```
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

## Output:

```
Hello, World!
```

# Parts of the Program

## 1. Comment:

- Single-line comments use `//`, while multi-line comments use `/* ... */`.

```
// This is a single-line comment

/*
   This is a
   multi-line comment
*/
```

## 2. Header File:

- Header files like `stdio.h` are required to use functions such as `printf()` or `scanf()`.

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

## 3. Main Function:

- The `main()` function is the program's entry point.
- `int main()` indicates that the function returns an integer (0 for success, 1 or more for failure).

```
int main() {
    // Program code
    return 0; // Indicates successful program execution
}
```

## 4. The `printf()` Function:

- This function is used to print output to the screen.
- `\n` is an escape sequence meaning newline (new row).

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

---

# VARIABLES AND DATA TYPES

Variables are "containers" for storing values. The data type determines the kind of value that can be stored in a variable.

# Types of Data in C

1. **int** - For integer values.

```
int number1; // Variable without initialization (random value)
int number2 = 20; // Variable initialized with value 20
```

2. **float** - For decimal values.

```
float decimal = 3.14;
```

3. **char** - For storing a single character.

```
char letter = 'A';
```

Here is a complete diagram of data types in C:

Alt text: A diagram showing the hierarchy of data types in C, starting from 'All text' and branching into 'Integer', 'Floating point', and 'Character'. The 'Integer' branch further splits into 'int', 'short', and 'long'. The 'Floating point' branch splits into 'float' and 'double'. The 'Character' branch splits into 'char' and 'wchar\_t'. The 'Integer' branch also includes a note 'Not found or type unknown'.

## Naming Variables

- Variable names must start with a letter or an underscore (`_`).
- Spaces or punctuation marks (such as `?`, `!`, etc.) are not allowed.
- Case-sensitive: `name` and `Name` are different variables.

Example:

```
int age = 20;    // Valid
float height = 170; // Valid
char initial = 'A'; // Valid
int ageOfFather = 45; // Valid

int 2age = 20;    // Invalid (cannot start with a number)
```

## Complete Example

Here is an example program using variables and data types:

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

```
int main() {  
    int age = 25;  
    float height = 170.5;  
    char initial = 'A';  
  
    printf("Age: %d years\n", age);  
    printf("Height: %.2f cm\n", height);  
    printf("Initial: %c\n", initial);  
  
    return 0;  
}
```

### Output:

```
Age: 25 years  
Height: 170.50 cm  
Initial: A
```

---

Revision #2

Created 7 February 2025 05:48:55 by BH

Updated 7 February 2025 16:08:08 by BH