C PROGRAMMING HANDBOOK BY SAICHANDAN GORLI

CHAPTER 1 : VARIABLES, CONSTANTS & KEYWORDS

VARIABLES

A variable is a container which stores a 'value'.In kitchen, we have containers storing. Rice, Dal, Sugar etc. Similar to that. variables in C stores value of a constant.

Example:

```
a = 3; // a is assigned "3"
b = 4.7; // b is assigned "4.7"
c = 'A'; // c is assigned 'A'
```

DATA TYPES AND SIZEOF

1. int → 4 byte (32 bits)
2. char → 1 byte (8 bits)
3. short → 2 byte (16 bits)
4. long → 4 byte (32 bits)
5. float → 4 byte (32 bits)
6. double → 8 byte (64 bits)
7. long long → 8 byte (64 bits)
8. long double → 8 byte (64 bits) or 12 or 16 bytes **depending on compiler and platform**
9. Bool → 1 byte (8 bits)

RULES FOR NAMING VARIABLES IN C

1. First character must be an alphabet or underscore(_).
2. No commas, blanks are allowed.
3. No special symbols other than (_) allowed.
4. Variables names are case sensitive

We must create meaningful variable names in our programs. This Enhances readability of our program.

CONSTANTS

An entity whose value does not change is called as a constant.

A variable is an entity whose value can be changed.

TYPES OF CONSTANTS

Primarily, there are three types of constants:

1. Integers Constants"color:hotpink;">→ 1,2,3,4
2. Real Constants"color:hotpink;">→ 322.1,2.5,7.0

3. Character Constants → 'a','$','@' (must be enclosed within single inverted commmas)

KEYWORDS

Thes are reserved words, whose meaning is already known to the compiler. There are 32 keywords available in C.

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

OUR FIRST C PROGRAM

```
#include <stdio.h>


int main()
{
    printf("hello world");
    return 0;
}
```

BASIC STRUCTURE OF A C PROGRAM

All C programs have to follow a basic stucture. A C program starts with a main function and execute instructions present inside it.

Each instructions is terminated with a semicolon (;).

There are some rules which are applicable to al the C program:

1. Every program's excution starts from main() function.
2. All the statements are terminated with semicolon.
3. Instructions are case-sensitive.
4. Instructions are excuted in the same order in which they are written.

COMMENTS

Comments are used to clarify something about the program in plain language, It is a way for us to add notes t our program. There are two types of comments in C.

1. Single line Comment: Single-line comments starts with two forward slahes(//). Any information after the slashes // lying on the same line would be ignored (will not be excuted).

```
// This is Single line comment.
```

2. Multi-line Comment: A multi-line comment starts with /_ and ends with */. Any information between /* and _/ will be ignored by the compiler.

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

Note : Comments in a C program is not excutd and are ignored.

COMPILER AND EXCUTION

A compiler is a computer program which converts a C program into machine language so that it can be easily understood by the computer.

A C program is written in plain text.

This plan text is combination of instruction in a particular sequence. The compiler performs some basic checks and finally converts the program into an excutable.

LIBRARY FUNCTION

C language has a lot of valuable library function which is used to carry out certain tasks. For instance printf() function is used to print values on the screen.

```
#include<stdio.h>

int main()
{
    int a = 6;
    printf("The output of the program is %d", a);
    //%d for integers
    //%f for reak values (floating-point numbers)
    //%c for characters

    return 0;
}
```

TYPES OF VARIABLES

1. Integers vairables → int a = 3;
2. Real variables → int a = 7; float b = 7.7;
3. Character variables → char a = 'b';

RECEIVING INPUT FROM THE USER

In order to take input from the user and assign it to a variable, we use ***scanf()*** function

Syntax:

```
scanf("%d", &a);
```

'&' is the "address of" operator and it means that the supplied value should be copied to the address which is indicated by variable a.