CHAPTER 2

CONSTANTS, VARIABLES AND DATATYPES

CHARACTER SET:

The characters that can be used to form words, numbers and expressions depend upon the computer on which the program is run.
The characters in C are grouped into the following categories;

1. **Letters** **(like: A-Z or a-z)**
2. **Digits** **(like:0-9)**
3. **Special characters** **(, . ; ? : $ # etc)**
4. **White spaces** **(like: blank spaces, tab, carriage return, new line)**

C TOKENS:

The smallest individual units in a program are known as Tokens.
They are,

1. **Keywords**
2. **Identifiers**
3. **Constants**
4. **Strings**
5. **Special Symbols**
6. **Operators**

I) Keywords:

They are explicitly reserved identifiers & cannot be used as names for the program variables or other user-defined program elements.
There are 32 keywords in C.
Ex: - break, case, char, const, etc.

II) Identifiers:

Identifiers refers to the names of variables, functions, arrays created by the programmer. They are fundamental requirements of any language.

Rules:

***) Only alphabetic characters, digits and underscores are permitted.**
***) The name can't start with a digit.**
***) Uppercase & Lowercase letters are distinct.**
***) A declared keyword can't be used as a variable name.**

III) Constants:

Constants refer to a fixed value that do not change during the execution of a program.
Ex: 1, 1000, 1.00, etc.

IV) Strings:

A string is a sequence of characters enclosed in a "double quotes".
Ex: "HELLO" , "2002", "x", etc.

V) Special Symbols:

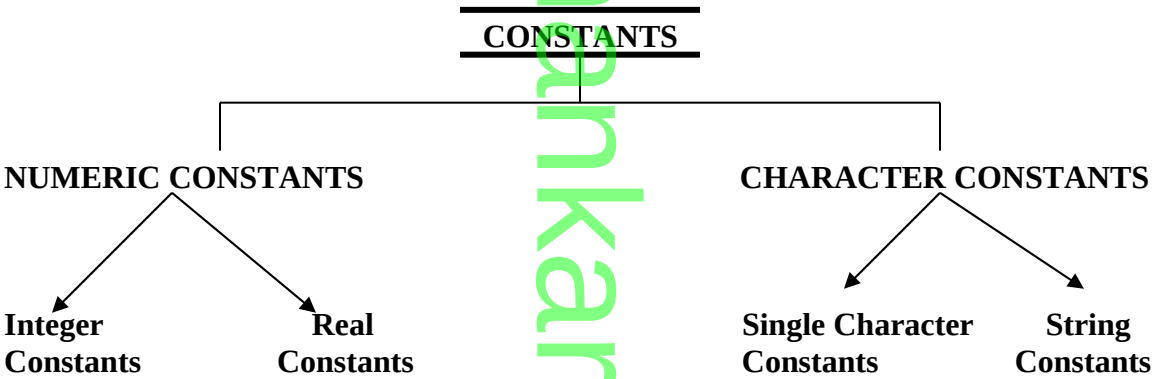
These are used in arrays, opening & closing of the program or a function.
Ex: [ ] , { }

VI) Operators:

Operators are those, which operate on the operand.
Ex: +, -, /, etc.

CONSTANTS

Constants refer to a fixed value that do not change during the execution of a program.



VARIABLES

DEFINATION:

A *variable* is a data name that may be used to store a data value. A *variable* may take different values at different times during the execution of a program.

Declaration of Variables:

All variables must first be declared before they are used in programs. This helps the compiler to do two things;

- **i) Reserve the amount of memory required.**
- **ii) Correctness of the objects use**

If there is an undeclared object is encountered in the program, then error is generated with the diagnostic message.

Rules

- **i) The name of a variable is a meaningful sequence of letters, digits and the underscore character.**
- **ii) The first character must begin with an alphabet or an underscore.**
- **iii) All succeeding characters must be either letters or digits.**

Ex:

```
Int I;
Char myname;
```

DATA TYPES

C language is rich in its *datatypes.* ANSI C supports the following data types.

1. **Primary (or Fundamental) data types.**
2. **User-defined data type.**
3. **Derived data type.**

I) Primary Datatype:

Name	Size(in bytes)	Range of values
a) char	1	-128 to 127
b) int	2	-32,768 to 32,767
c) float	4	3.4e-38 to 3.4e+38
d) double	8	1.7e-308 to 1.7e+308

II) User-Defined Datatype:

a) **structures**
b) **unions**
c) **enum**

III) Derived Datatype:

a) **array**
b) **function**
c) **pointer**

Modifiers:

The basic datatypes may have several *modifiers* preceding them to serve the needs of various situations. They are,

a) **signed**
b) **unsigned**
c) **long**
d) **short**

may be applied to character and integer datatypes. However, the modifier *long* may also be applied to *double*.

After applying the modifiers to basic datatypes, their size is as follows,

Name	Size(in bytes)	Range of values
a) unsigned char	1	0 to 255
b) signed char	1	-128 to 127
c) unsigned int	2	0 to 65535
d) signed int	2	-31768 to 32767
e) short int	2	-31768 to 32767
f) long int	4	----------
g) long double	10	----------

END OF CHAPTER 2