Parul® University

Fundamental of Programming

Prof. Umang Panchal, Asst. professor
Computer Science & Engineering

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-3

Constants, variables and data types

Introducing C

- C program is developed by Dennis Ritchie in 1972 at Bell laboratory in USA.
- Designed for systems programming
  - Operating systems
  - Utility programs
  - Compilers
  - Filters
- Evolved from B, which evolved from BCPL

Why C language is used??

- Currently, the most commonly-used language for embedded systems is C language.
- Very portable: compilers exist for virtually every processor
- Easy-to-understand compilation
- Produces efficient code

Features of C

Portability:

C Programs are portable i.e they can be run on any Compiler with Little or no Modification.

Powerful:

Provides Wide variety of '**Data Types**'
Provides Wide variety of 'Functions'
Provides useful Control & Loop Control Statements

Bit Manipulation :

C Programs can be manipulated using bits. We can perform different operations at bit level. We can manage memory representation at bit level.

Features of C

Effective use of pointers:

Pointers has direct access to memory.

C Supports efficient use of pointer

Structured programming language

C is a structured programming language in the sense that **we can break the program into parts using functions**. So, it is easy to understand and modify.

Memory Management

It supports the feature of **dynamic memory allocation**. In C language, we can free the allocated memory at any time by calling the **free()** function.

C Tokens

C tokens are the basic buildings blocks in C language which are constructed together to write a C program.

Each and every smallest individual units in a C program are known as C tokens.

The keywords

- "**Keywords**" are words that have special meaning to the **C** compiler.
- Their meaning cannot be changed at any instance.
- Serve as basic building blocks for program statements.
- All keywords are written in **only lowercase.**

Keywords in ANSI *C*

auto
break
case
char
const
continue
default
do

double
else
enum
etern
float
for
goto
if

register
return
short
signed
sizeof
static
struct
int

switch
typedef
union
unsigned
void
volatile
while
long

The identifiers

- Identifier are created to give unique name to C entities to identify it during the execution of program.

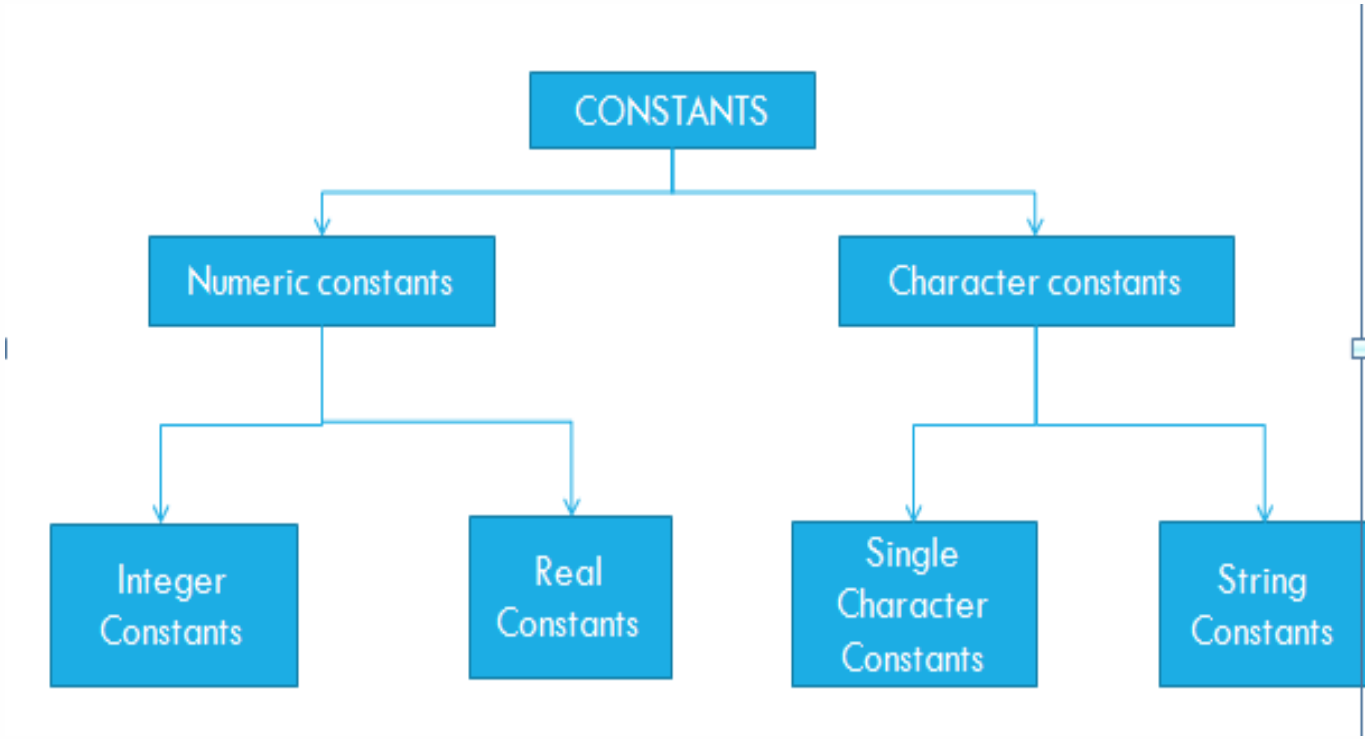
- Cannot use C keywords as identifiers
- Must begin with alpha character or _, followed by alpha, numeric, or _
- Upper- and lower-case characters are important (case-sensitive)
- Must consist of only letters, digits or underscore (_).
- Only first 31 characters are significant.
- Must NOT contain spaces ( ).

EXAMPLES

IDENTIFIER	VALID?	REASON IF INVALID
`totalSales`	Yes	
`total_Sales`	Yes	
`total.Sales`	No	Cannot contain .
`4thQtrSales`	No	Cannot begin with digit
`totalSale$`	No	Cannot contain $

Constants

Constants in C are the fixed values that do not change during the execution of a program.



Program structure

A sample C Program

```
#include<stdio.h>
#include<conio.h>
Void main()
{
        --other statements
}
```

Header Files

- The files that are specified in the include section is called as header file
- These are precompiled files that has some functions defined in them
- We can call those functions in our program by supplying parameters
- Header file is given an extension **.h**
- C Source file is given an extension **.c**

Main function

- This is the entry point of a program
- When a file is executed, the start point is the main function
- From main function the flow goes as per the programmers choice.
- There may or may not be other functions written by user in a program
- Main function is compulsory for any C program

Running a ‘C’ Program

- **ANSI C supports three classes of data types**.

1. Primary or Fundamental data types.
2. User-defined data types.
3. Derived data types.

DATA TYPES

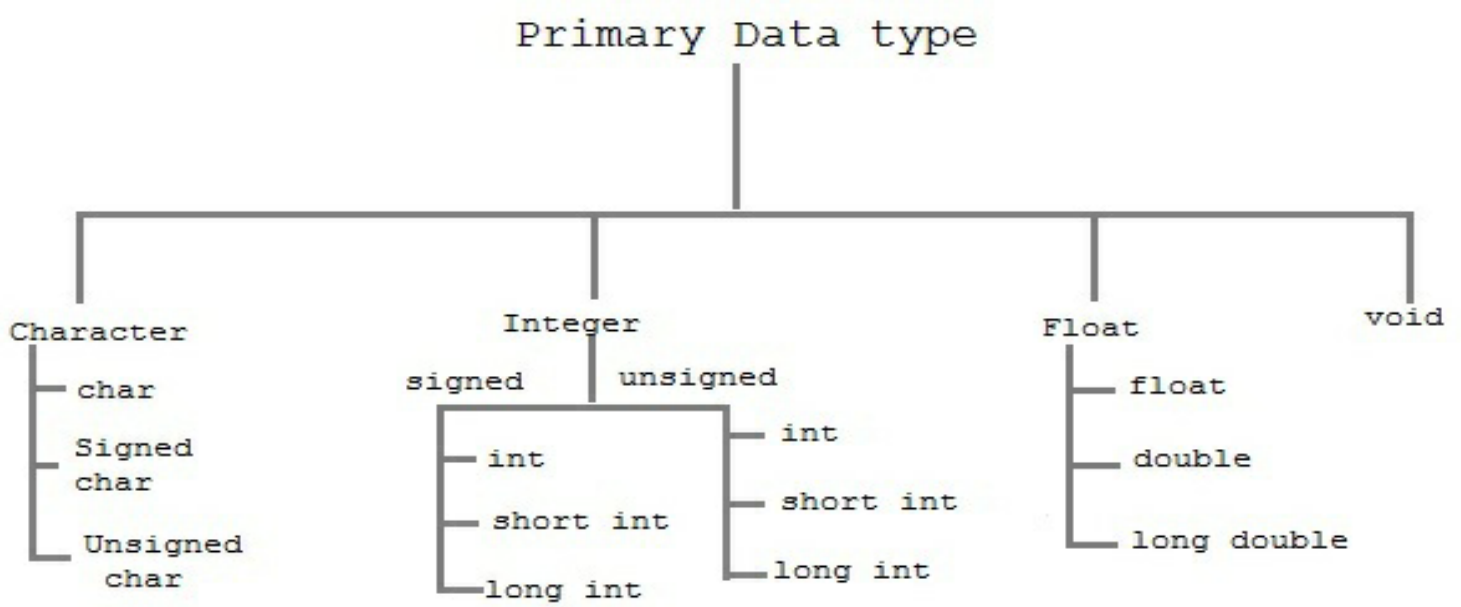
- A data type is
  - A set of values AND
  - A set of operations on those values
- A data type is used to
  - Identify the type of a variable when the variable is declared
  - Identify the type of the return value of a function
  - Identify the type of a parameter expected by a function

DATA TYPES

- **ANSI C supports three classes of data types**.

1. Primary or Fundamental data types.
2. User-defined data types.
3. Derived data types.

Primary Data Types in C



Integer Types

Size and Range Of Data types on 16 bit machine

Type	Bytes	Values
int	2 or 4	-32, 768 to 32, 767
unsigned int	2 or 4	0 to 65, 535
signed int	2 or 4	-32, 767 to 32, 767
short int	2	-32, 767 to 32,767
unsinged short int	2	0 to 65, 535
signed short int	2	-32, 767 to 32, 767
long int	4	-2,147,483,647 to 2,147,483,647
signed long int	4	-2,147,483,647 to 2,147,483,647
unsigned long int	4	0 to 4, 294,967,294

Floating Point Types

DATA TYPE	SIZE	RANGE
Float	4 bytes	3.4e - 38 to 3.4e + 38
Double	8 bytes	1.7e – 308 to 1.7e + 308
Long double	10 bytes	3.4e – 4932 to 1.1e + 4932

User-defined type declaration

- C allows user to define an identifier that would represent an existing **data type.**
- The general form is **typedef type identifier;**

Eg:

typedef int units;

typedef float marks;

- Another user defined data types is enumerated data type which can be used to declare variables that can have one of the values enclosed within the braces.
- **enum identifier {value1,value2,……valuen};**

Parul® University

Derived data type

- C allows a different types of derived data structure
- Different types of datatypes are
- array
- Functions
- Pointer
- Structure

DECLARATION OF VARIABLES

- **Declarations does two things:**

- It tells the compiler what the variable name is
- It specifies what type of data the variable will hold

- **Primary Type Declaration**

- The syntax is
- **Data-type v1,v2…..vn;**

Eg:

int count;

double ratio, total;

User-defined type declaration

- C allows user to define an identifier that would represent an existing **int data type.**
- The general form is **typedef type identifier;**

Eg:

typedef int units;

typedef float marks;

- Another user defined data types is enumerated data type which can be used to declare variables that can have one of the values enclosed within the braces.
- **enum identifier {value1,value2,……valuen};**

User-defined type declaration

Declaring a variable as constant

Eg:

const int class_size=40;

- This tells the compiler that the value of the int variable class_size must not be modified by the program.

Declaring a variable as volatile

- By declaring a variable as volatile, its value may be changed at any time by some external source.

Eg:

volatile int date;

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in