Data Types and Variables

1. INTRODUCTION
2. CHARACTER SET
3. C TOKENS
4. DATA TYPES

Introduction

- The Sequence of the Instruction written to perform specific task is called the Program.
- This Instructions are formed using special symbols and words according to some rigid rule is known as syntax rules.
- Every Instruction must be written according to the syntax rule of the languages.
- Like any other languages C languages has its vocabulary and grammar. They are follow up for making the program.

Character Set

- Character Set is the set of the character.
- This characters are used to form the word, numbers and expression.
- The characters in the c are grouped into the following categories.
  - Letters : Uppercase A…Z, Lowercase a…z
  - Digits : All decimal digits 0 to 9
  - Special Character : ,(comma) .(period) ; (semicolon) :(colon), & (ampersand), # (number sign) etc.
  - White spaces : Blank Space, Horizontal Space, New Line.

C Tokens

- Smallest Individual units are known as C Tokens.
- There are six types of the C Tokens.
  - Keywords
  - Identifiers
  - Constants
  - String
  - Special Symbols
  - Operators.

Keywords

- Every C word is classified as either keyword or identifier.
- All keywords have fixed meanings and you can not change its meanings.
- Keywords serve as a basic building blocks for program statement. ANSI C supports 32 keywords.
- Ex: int, float, double, extern, static, auto, continue, if, goto short, long etc. are the keywords

Identifiers

- Identifiers refer to the names of variables, functions and arrays.
- These are user defined names and consists of sequence of letters and digits.
- Both uppercase and lowercase letters are permitted to make the identifier but generally lowercase letters are used to make the variable.

Rules for Identifiers

- First character must be alphabet
- Must not contain white space
- Only first 31 characters are significant
- Can not use keyword as a identifier

Constants

- Constants referred as a fixed value that don't change during the execution of the program.
- C Support Several types of constants:
  - Numeric Constants
    - Integer constants
    - Real Constants
  - Character Constants
    - Single Character Constants
    - String Constants

Integer Constants

- An integer constants refers as a sequence of digits.
- There are three types of integer constants.
  - Decimal Integer
  - Octal Integer
  - Hexadecimal

Decimal Integer

- It consists of 0-9 digits, preceded by an optional – or + sign.
- Valid example of decimal integer are : 123 , -321, 0 , 654321, +78
- Embedded spaces, comma and non digit characters are not permitted between digits.
- 15 750, 20,000, $1000 are Illegal

Octal Integer

- An Octal Integers consists of digits 0-7 with a leading 0.
- Example: 037, 0, 0435, 0551

Hexadecimal Integer

- A sequence of digits preceded by 0x or OX is considered as Hexadecimal Integer.
- Hexadecimal Integer includes 0 to 9 digits and A to F letters.
- Example: 0x9, 0x9F, OXBC etc are valid.

Real Constants

- Integer numbers are inadequate to represent quantities such as distance, heights, temperature, price and so on. These quantities are represented by a number containing fractional parts like 12.32. Such numbers are called real constants
- These numbers having a whole number followed by a decimal digits.
- Example: 0.85, -0.75, 85.45, +241.54

- A real number may also be expressed in exponential (Scientific) notation.
- General form :mantissa e exponent
- The mantissa is either a real number or integer number.
- Exponent is an integer number with + or – sign.
- The letter e separating mantissa and exponent can be written either in lowercase or in uppercase letter.

- Example : 215.65 may be written as 2.1565e2 in exponential notation. e2 means multiple by 10^2
- 75000 written as 7.5E4 or 7.5E+4
- -0.00038 written as -3.8e-4
- Comma, White space and dollar space is not permitted in digits.
- 25,0.000 , 7.1 e 4 , 1.5 E 2.5 (exponent must be an integer),$255.

Single Character Constants

- A single character constant contains a single character enclosed with a pair of single quotation mark (‘ ’).
- Example: ‘5’, ‘x’, ‘;’ , ‘ ’

- Note that character constant ‘5’ is not same as number 5.
- Character constant have a integer value known as ASCII value

- Example :
- printf (“%d”, ‘a’) would print the number 97.
- printf(“%c”, ‘97’) would print the letter a.

String Constants

- A String Constant is a sequence of characters enclosed in double quotation mark.
- Example: “Hello” , “1987”, “Well Done”, “X”
- Note : A single String constant does not have an equivalent integer value, while a character constant has an equivalent integer value.

Backslash Character Constant:

- Backslash character constant are used in output function.
- Example: ‘\n’ new line,
  ‘\t’ horizontal tab,
  ‘\” single quote etc.

Variables

- A variable is a data name that may be used to store a data value.
- Unlike constants that remain unchanged during the execution of the program, a variable may take different value at different time.

Rules to declare the variables

- They must begin with letter or underscore(_)
- First character may be followed by letters or digits.
- Both lowercase and uppercase letters are distinct.

  Example: Total and total are not same
- It should not be a keyword
- White space, Dollar sign are not allowed.
  Example: John, delhi, First_tag, int_type ---- valid

  Price$, char, group one, 123, (area) ------- Invalid

Declaration of variables

- Syntax : data_type v1,v2,v3….vn ;

  where v1,v2..vn are different variable name.

- Example: int count;

  int number, total;

  double ratio;

  float price;

  char c;

  where int, double, float and char are data type.

Assigning Values to variables

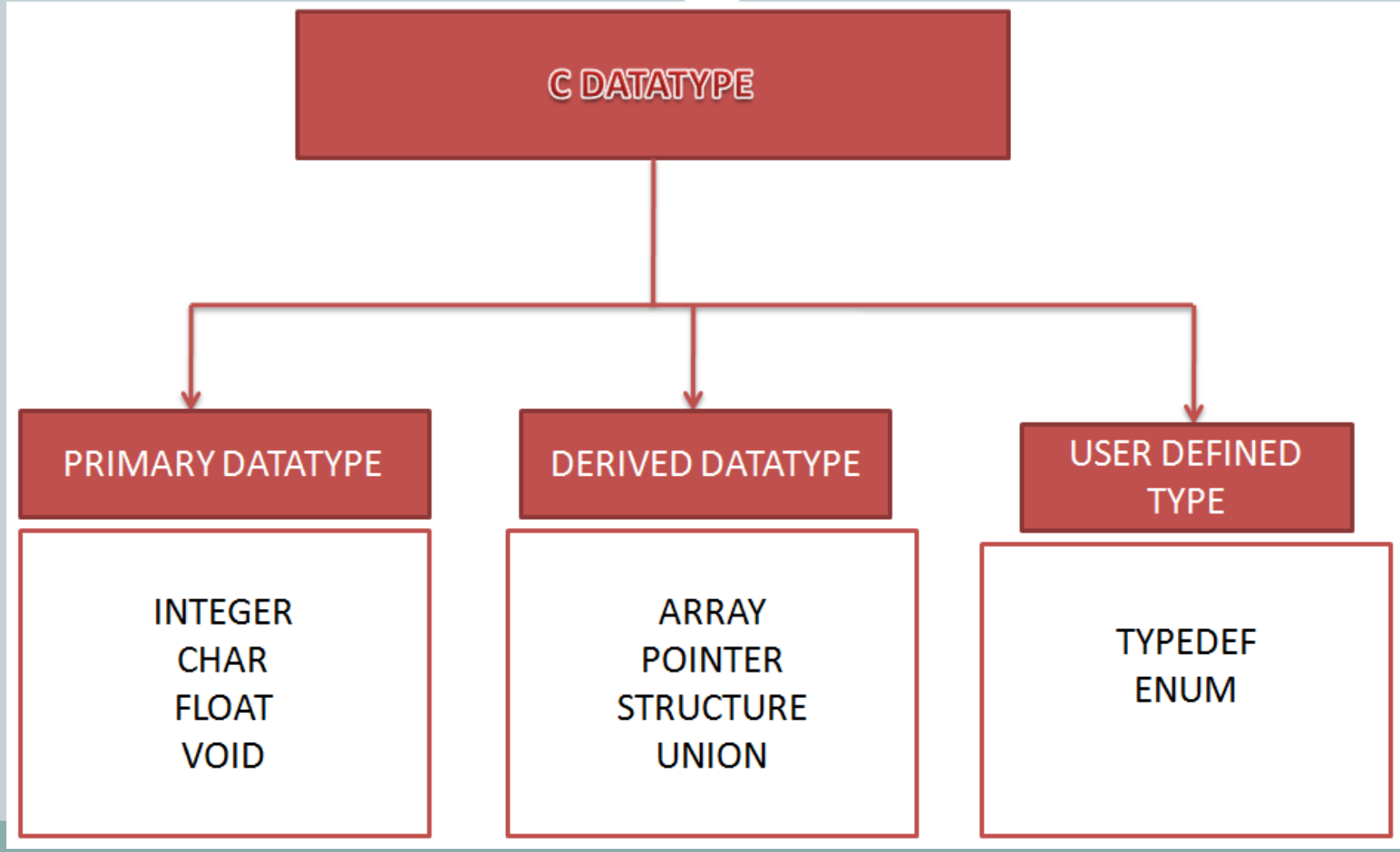
- Assignment operator (=) is used to assign value to a variable.
- Example: price = 12.50;

  ratio = 12.2345;

  number=12;

  c='a';

Data Types


C DATATYPE
PRIMARY DATATYPE
DERIVED DATATYPE
USER DEFINED TYPE
INTEGER
CHAR
FLOAT
VOID
ARRAY
POINTER
STRUCTURE
UNION
TYPEDEF
ENUM

Primary Data Types

Integral Type

Integer

Signed	Unsigned
int	unsigned int
short int	unsigned short int
Long int	unsigned Long int

Character

char

signed char

unsigned char

Floating Point Type

float	double	long double

void

Integer Types

- Integers are whole numbers with a range of values supported by a particular machine.
- There are signed integer and unsigned integer. Signed integer uses one bit for sign and other bits for magnitude of the number. Unsigned integers are always positive. It does not contain any bit for sign so that it occupies all the bit for the magnitude of the number.
- By using equation -2^n to +(2^n)-1 we can find out the range of the number. Where n is the number of bits.

Size and Range of Integer Data

Type	size(bits)	Range
int or signed int	16	-2^{15} to $2^{15}-1$
unsigned int	16	0 to $2^{16}-1$
short int or signed short int	8	-2^{7} to $2^{7}-1$
unsigned short int	8	0 to $2^{8}-1$
long int or signed long int	32	-2^{31} to $2^{31}-1$
unsigned long int	32	0 to $2^{32}-1$

Floating Point Types

- Floating point numbers are stored in 32 bits with 6 digits of precision.
- Floating point numbers are defined in C by the keyword float.
- When the accuracy provided by float is not sufficient double data type is used. It uses 64 bits giving a precision of 14 digits.
- When you want to extend more precision you can use the long double data type. It uses 80 bits

Size and Range of Floating Point Data

Type	Size(Bits)	Range
float	32	3.4E-38 to 3.4E+38
double	64	1.7E-308 to 1.7E+308
long double	80	3.4E-4932 to 1.1E+4932

Void Types

- Void type has no values.
- Void type does not return any values.
- These are used to specify the return values from the function when they don’t have any value to return

Character Types

- Character types data in C are defined by keyword char.

Type	Size (bits)	Range
char or signed char	**8**	**-2^7 to $2^7 - 1$**
unsigned char	**8**	**0 to $2^8 - 1$**

Typedef

- Using typedef keyword we can define our own user defined data type.
- Syntax:

  typedef type identifier;

- Example:

  typedef int marks;

  marks sub1,sub2;

Enumeration

- Declare using keyword enum
- Syntax:

  enum identifier {value1, value2, value3,….,valueN};

- Example:

  enum day {Monday, Tuesday,……..,Sunday};