COMPUTATIONAL THINKING FOR STRUCTURED DESIGN

CHAPTER-2

DATA TYPES,USER I/O AND OPERATORS

DATA TYPES

- The data type specify the type of data that a variable can store.
- 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

C provides 5 primary or fundamental data types

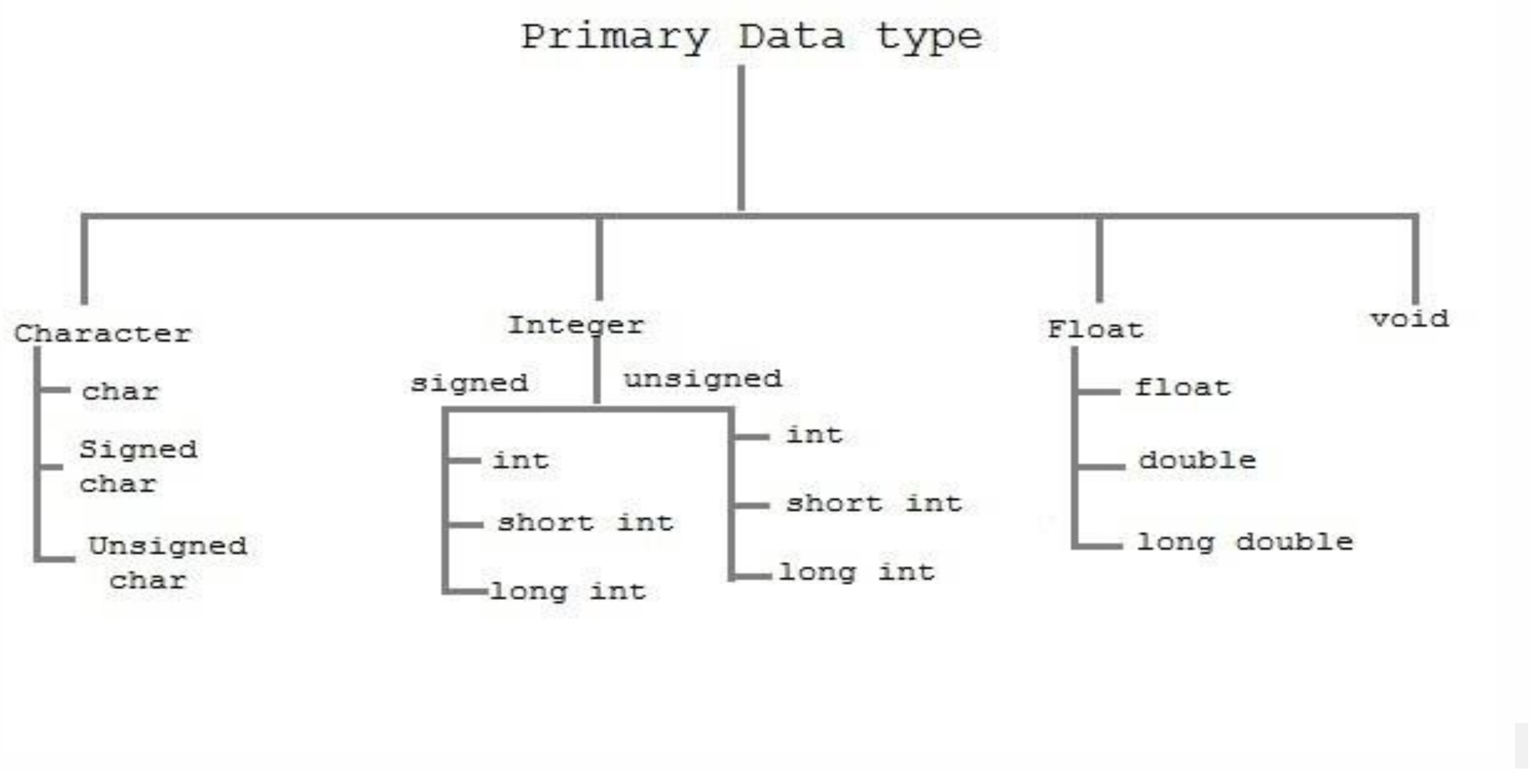
1. Character- char
2. integer- int
3. floating point -float
4. double
5. void.

Extended data type

We can also use the short, long, signed and unsigned keywords to extend the primary data types.

So, they are called extended data types

Primary Data Types in C

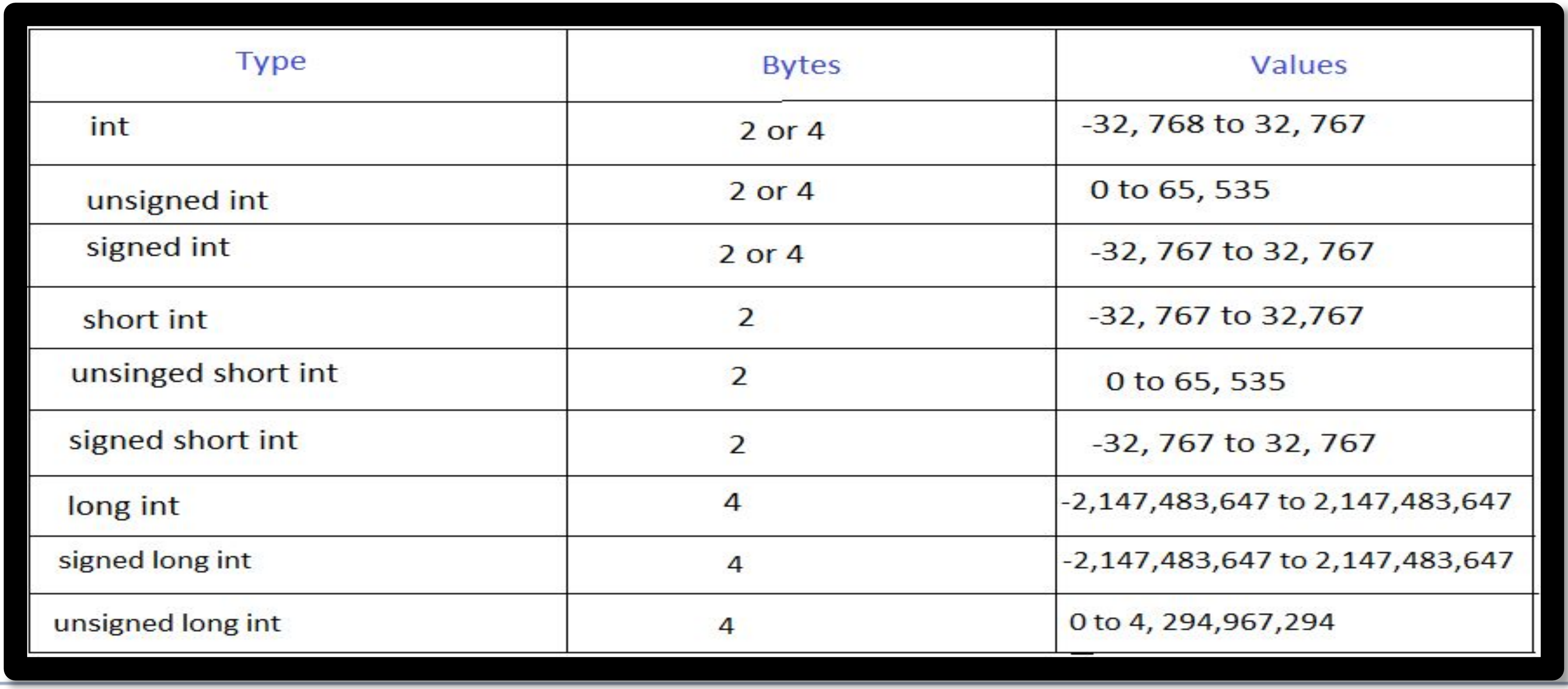


Primary Data type

- Character
  - char
  - Signed char
  - Unsigned char
- Integer
  - signed
    - int
    - short int
    - long int
  - unsigned
    - int
    - short int
    - long int
- Float
  - float
  - double
  - long double
- void

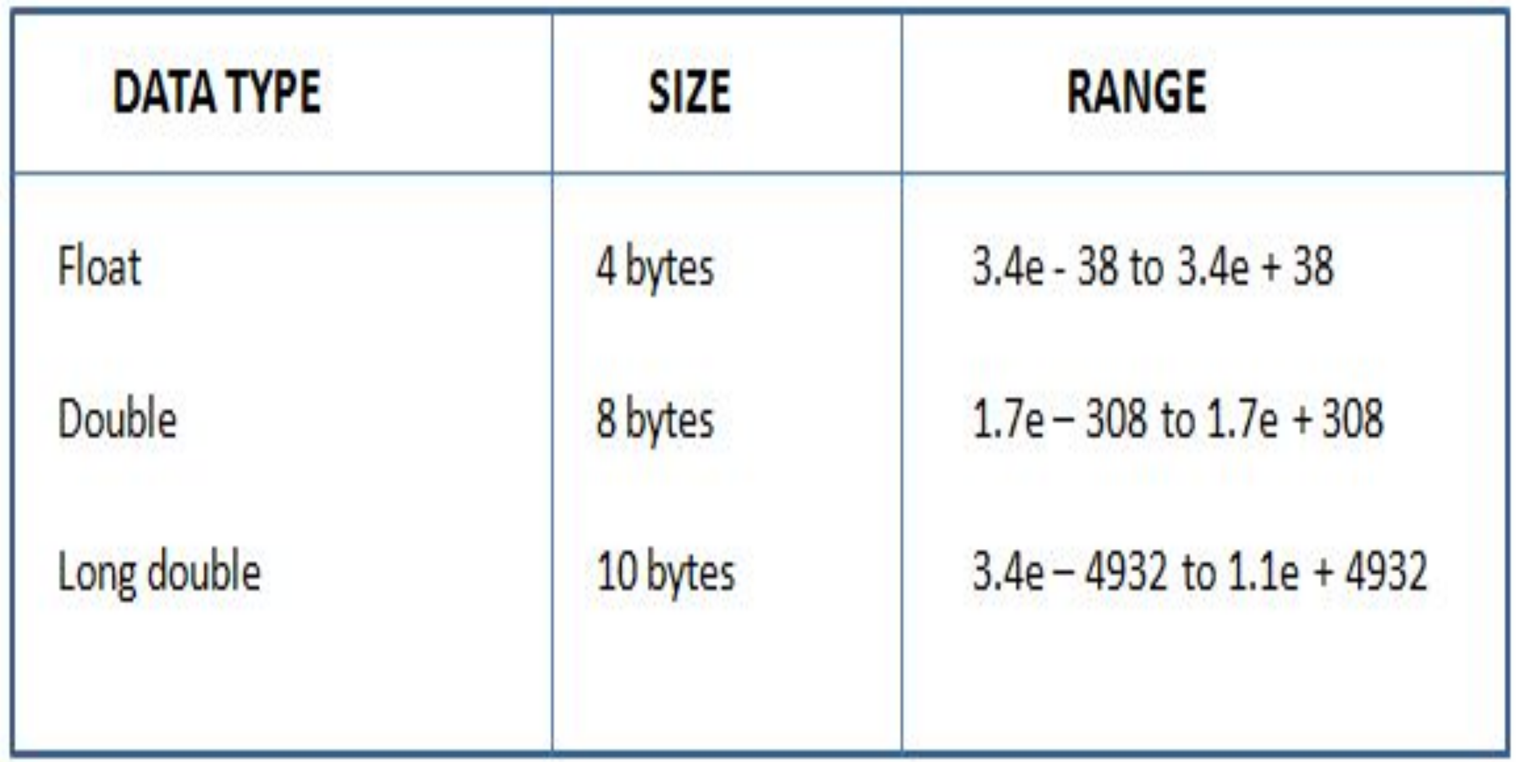
Integer Types

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

VOID

The void data type is generally used with function to denote that function is return nothing

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};**

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;
```

USER I/O

C language has standard libraries that allow input and output in a program. The **stdio.h** or **standard input output library** in C that has methods for input and output.

scanf() function

The scanf() method, in C, reads the value from the console as per the type specified.

Syntax:

scanf(“%X”, &variableOfXType); where **%X** is the format specifier in C.

Printf()function

The printf() method, in C, prints the value passed as the parameter to it, on the console screen.

Syntax:

printf("%X", variableOfXType); *where* ***%X*** *is the format specifier in C*

Input /output for basic datatype

The Syntax for input and output for these are:

- **Integer:**

  Input: scanf("%d", &intVariable); **Output:** printf("%d", intVariable);

- **Float:**

  Input: scanf("%f", &floatVariable); **Output:** printf("%f", floatVariable);

- **Character:**

  Input: scanf("%c", &charVariable); **Output:** printf("%c", charVariable);

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in