# Data-types in C

# Objectives of this session

- To learn about basic data types in C


- To learn how use declare data types in C

# Learning outcomes

- At the end of this session you will learn
  - About basic data types in C
  - How to declare the basic data types
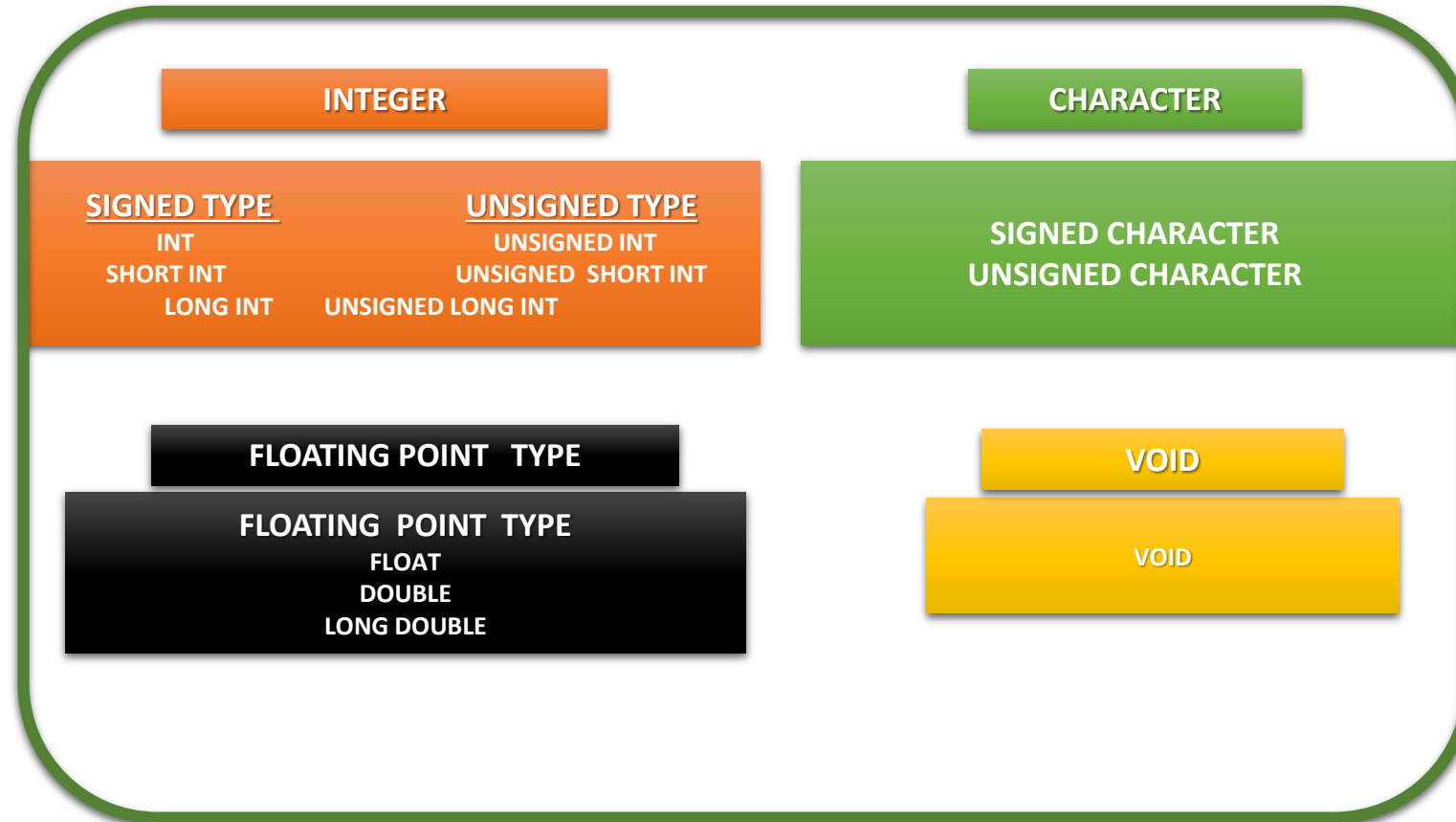
# Big Picture

- Processor works with <u>finite-sized</u> data
- <u>All data</u> implemented as a sequence of *bits*
  - Bit = 0 or 1
  - Represents the level of an electrical charge


- *Byte* = 8 bits


- *Word* = largest data size handled by processor
  - 32 bits on most older computers
  - 64 bits on most new computers

# Primary (built-in or Basic)Data types



| INTEGER | | CHARACTER |
|---|---|---|
| **SIGNED TYPE**     **UNSIGNED TYPE** | | **SIGNED CHARACTER** |
| INT    UNSIGNED INT | | **UNSIGNED CHARACTER** |
| SHORT INT    UNSIGNED SHORT INT | | |
| LONG INT    UNSIGNED LONG INT | | |

**FLOATING POINT   TYPE**

**FLOATING  POINT  TYPE**
FLOAT
DOUBLE
LONG DOUBLE

**VOID**

**VOID**

# Data types

**Basic data types: int, float, double, char, and void.**

✓**int:** can be used to store integer numbers (values with no decimal places).

✓**float:** can be used for storing floating-point numbers (values containing decimal places).

✓**double:** the same as type float, and roughly twice the size of float.

✓**char:** can be used to store a single character, such as the letter *a*, the digit character *6*, or a semicolon.

✓ **void:** is used to denote nothing or empty.

# Variables - Examples

int a;                          // declaring a variable of type int

int sum, a1, a2;         // declaring 3 variables
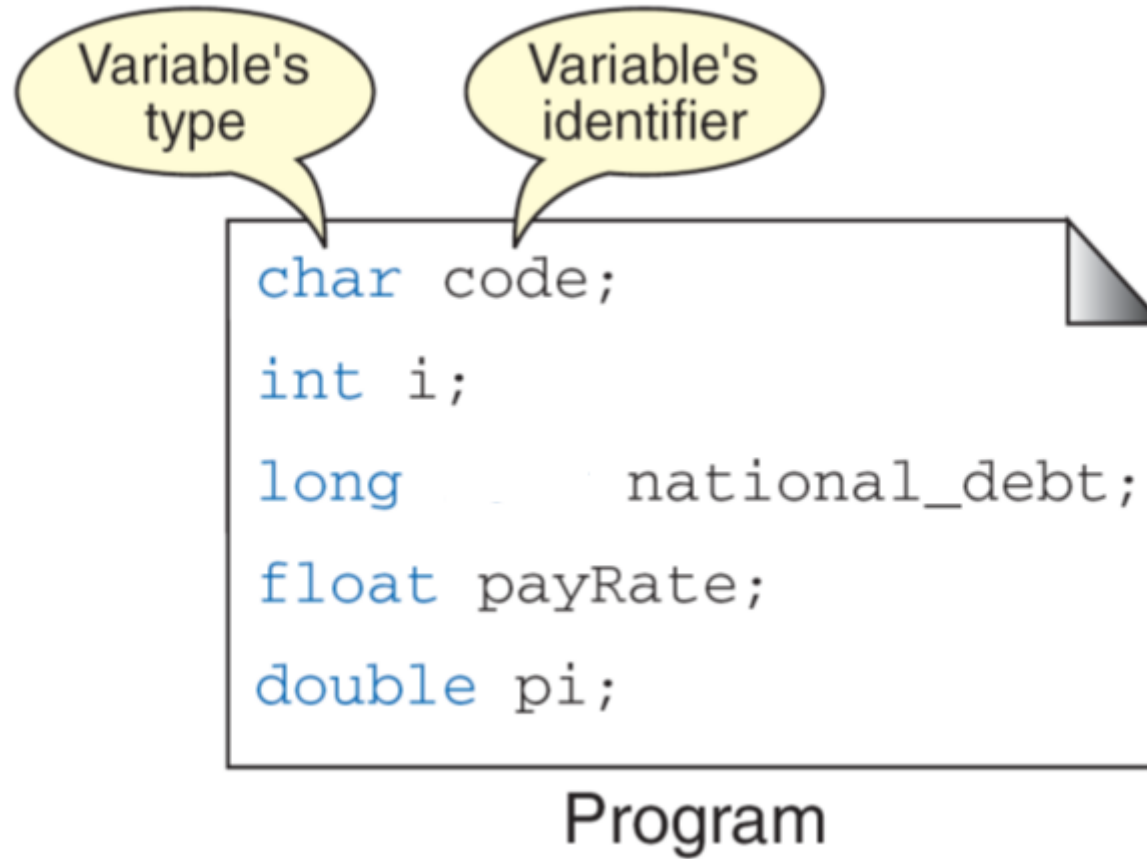
int x = 7;                      // declaring and initializing a variable

a = 5;                          // assigning to variable a the value 5

a1 = a;                        // assigning to variable a1 the value of  a
*L-value*      *R-value*

```
a1=a1+1;  // assigning to variable a1 the value of a1+1
          // (increasing value of a1 with 1)
```

# Variables -Example

# Integer Types

➢ The basic integer type is **int**

- ▪ The size of an **int** depends on the machine and on PCs it is normally 16 or 32 or 64 bits.

➢ **modifiers (type specifiers)**

- ▪ **short**:    typically uses less bits
- ▪ **long**:     typically uses more bits
- ▪ **Signed:**   both negative and positive numbers
- ▪ **Unsigned:**  only positive numbers

# SIZE AND RANGE OF VALUES FOR A 16-BIT MACHINE (INTEGER TYPE)

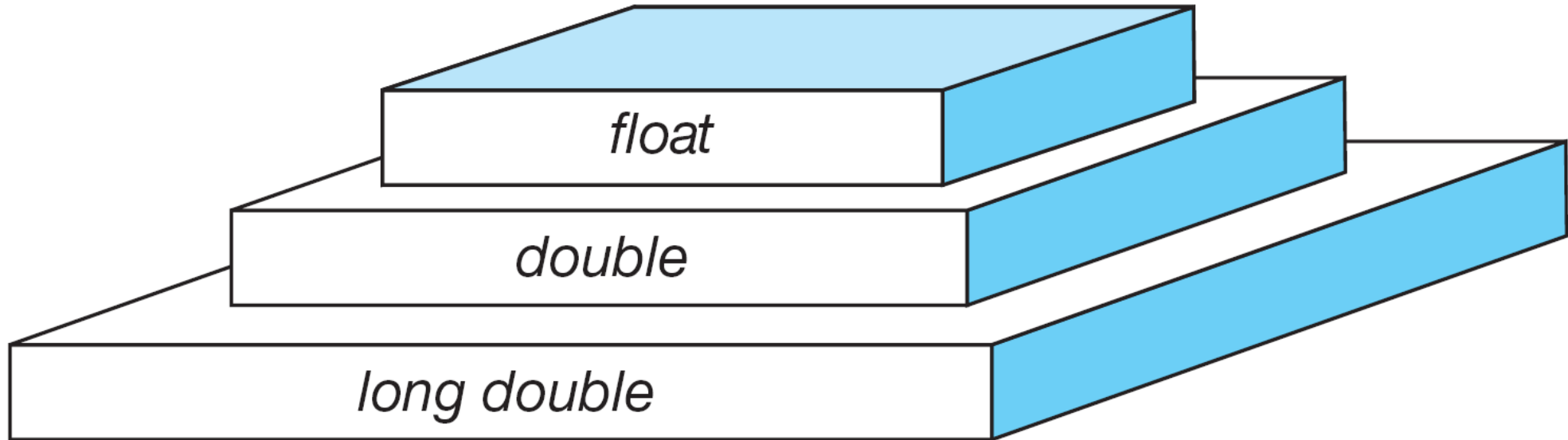| | Type | Size | Range |
|---|---|---|---|
| short | short int  or signed short int | 8 | -128 to 127 |
| | unsigned int | 8 | 0 to 255 |
| Integer | int or signed int | 16 | -32,768 to 32,767 |
| | unsigned int | 16 | 0 to 65,535 |
| Long | long int or signed long int | 32 | -2,147,483,648 to 2,147,483,647 |
| | unsigned long int | 32 | 0 to 4,294,967,295 |

# Data- Types in C
# Integer Data type
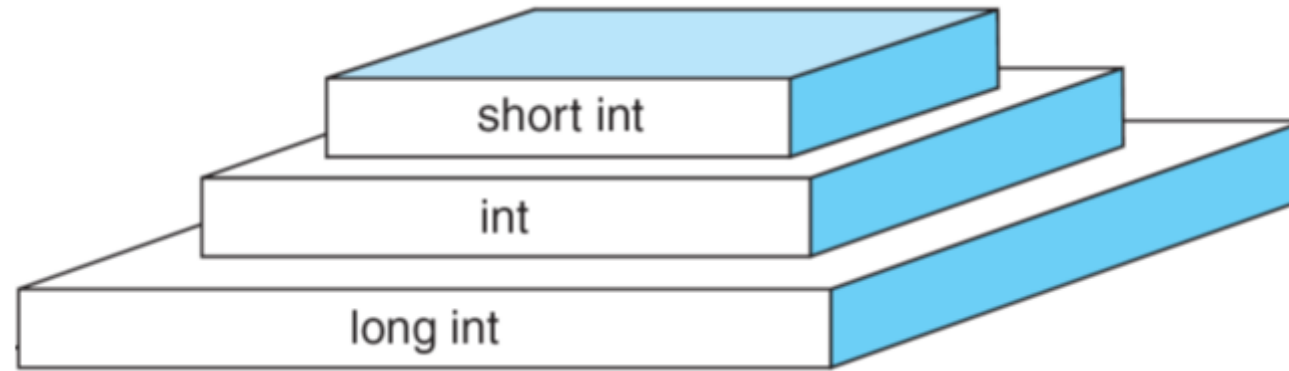
# Data- Types in C
# Float data type

# Summary

- The basic data types in C are
  - int
  - float
  - char
  - Double

- Size of the data types in machine dependent

- If we want a big integer to be stored , we can make use of keyword long[ long int ]

- Similarly, if we want higher precision to be achieved from a computation involving real numbers , then use long double

Go to posts/chat box for the link to the question
**submit your solution in next 2 minutes**
**The session will resume in 3 minutes**