SUNBEAM

C Programming

Trainer: Ketan G Kore

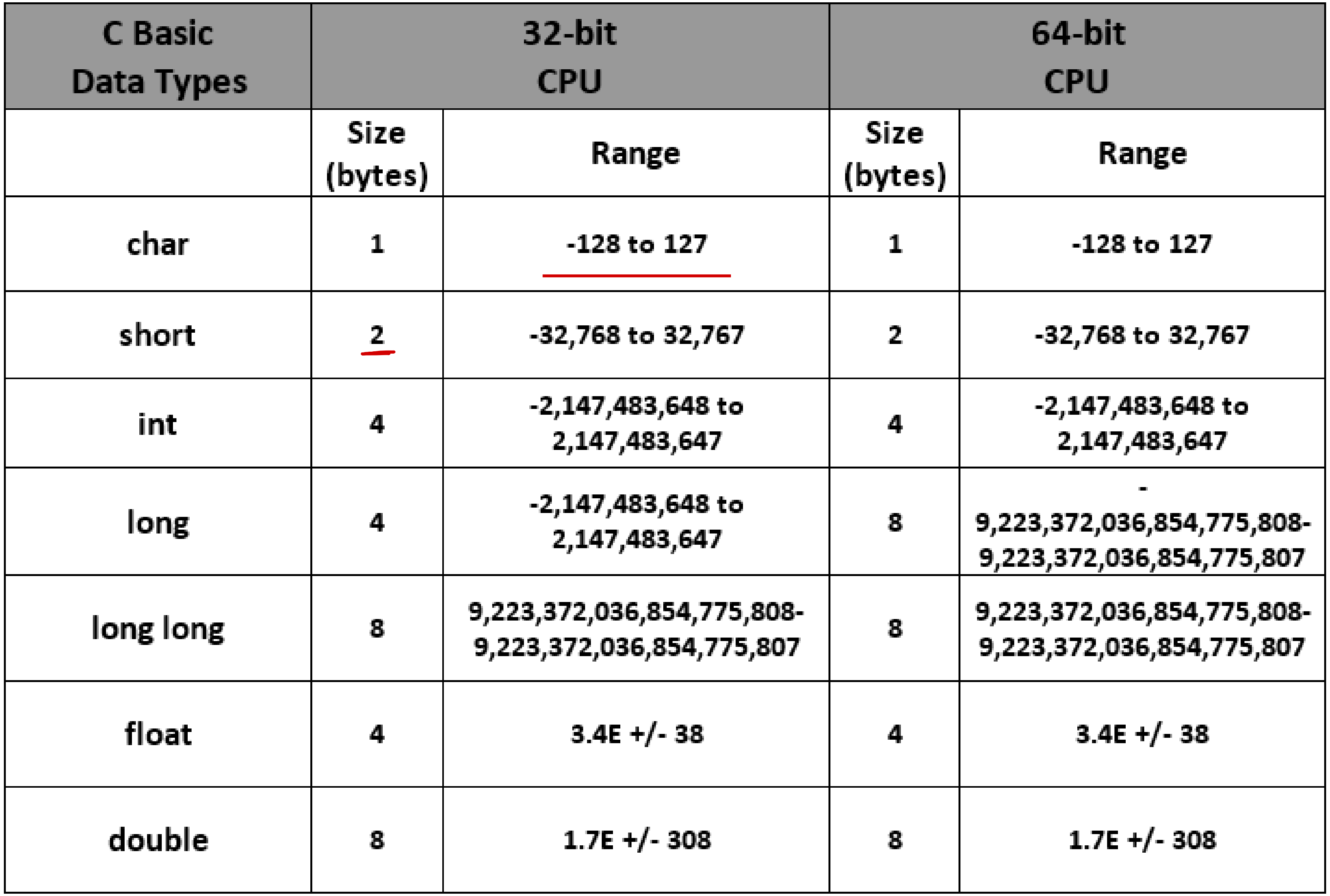
Data Types, Variables & Constants

- Variable examples
  - int number = 10;
  - double basic_salary = 20000.0;
  - char letter = 'A';
  - int roll_number;
  - roll_number = 20;
  - double price = 200.0;
  - price = 300.0;
- Constant examples
  - -23, 1L, 34U, 3UL, 0x41, 0101,
  - 1.234f, 1.234567e+2, …
  - "SunBeam", "A\101\x41"
- Each variable is assigned some memory location.
- Size of data type of given variable or constant is found by sizeof() operator.

Data Types

- Data type defines storage space and format of variable.
- Primitive types
  - int
  - short
  - long
  - char
  - float
  - double
- Integer types can be signed/unsigned
- Derived types
  - Array
  - Pointer
  - Function
- Type qualifiers
  - const and volatile
- printf() format specifiers
  - %d, %u, %o, %x
  - %hd, %hu
  - %ld, %lu
  - %c
  - %f, %e
  - %lf
- User defined types
  - struct
  - union
  - enum
- void type – represent no value.

Data Types



C Basic Data Types	32-bit CPU		64-bit CPU	
	Size (bytes)	Range	Size (bytes)	Range
char	1	-128 to 127	1	-128 to 127
short	2	-32,768 to 32,767	2	-32,768 to 32,767
int	4	-2,147,483,648 to 2,147,483,647	4	-2,147,483,648 to 2,147,483,647
long	4	-2,147,483,648 to 2,147,483,647	8	- 9,223,372,036,854,775,808- 9,223,372,036,854,775,807
long long	8	9,223,372,036,854,775,808- 9,223,372,036,854,775,807	8	9,223,372,036,854,775,808- 9,223,372,036,854,775,807
float	4	3.4E +/- 38	4	3.4E +/- 38
double	8	1.7E +/- 308	8	1.7E +/- 308

printf() and scanf()

- #include <stdio.h> -- function declaration

- printf()
  - Used to print values & string on terminal.
  - Various format specifiers %d, %c, %f, …
  - Formatting: %5d, %-7d, %08d, %8.2f, …

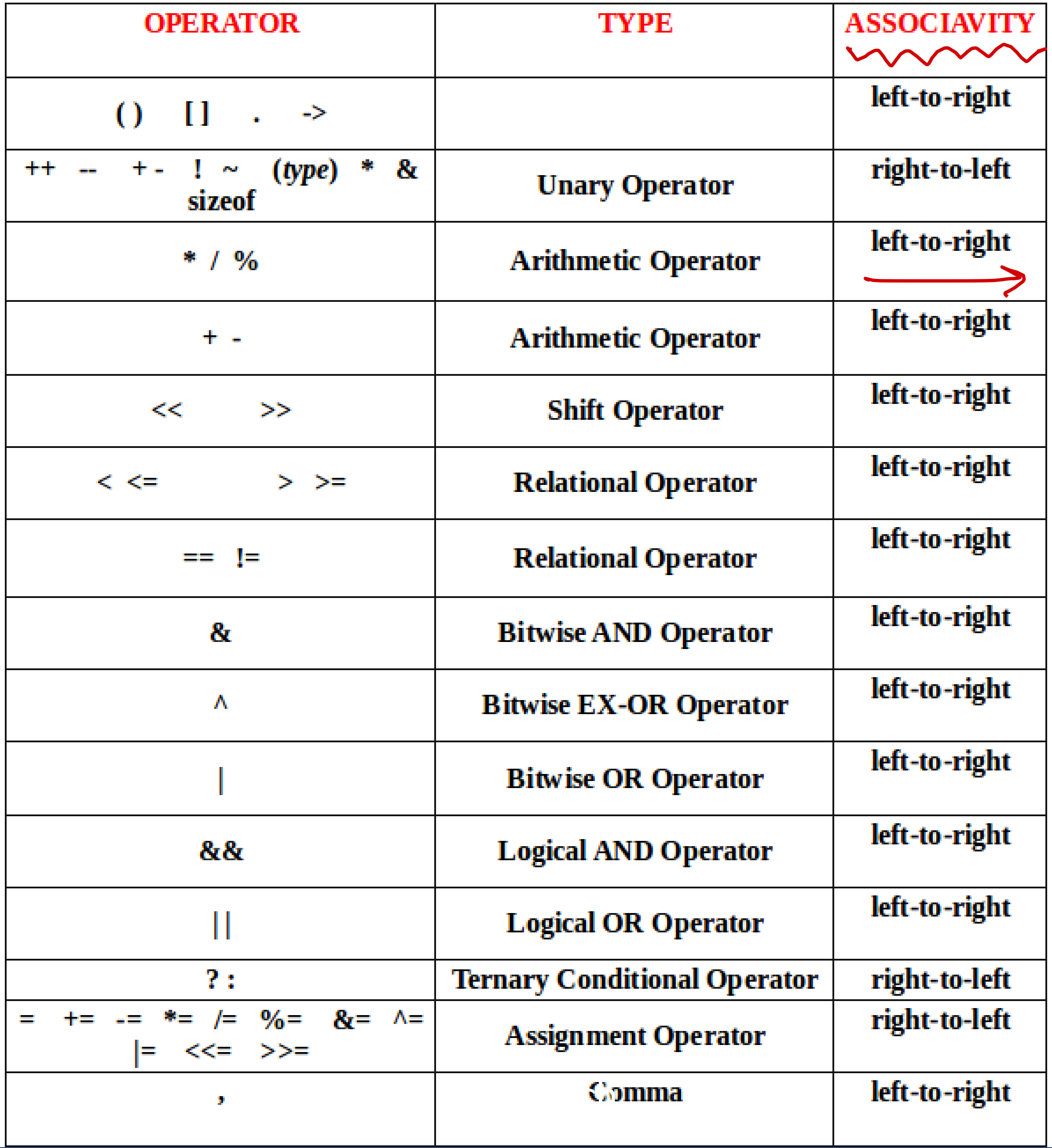
- scanf()
  - Used to input values from user.
  - Same format specifiers as of printf().
  - Do not use any char other than format specifiers in format string.
  - To skip a char from input use %*c.

Operators

- Types of operators
  - Arithmetic Operators (+ , - , * , / , %)
  - Assignment & shorthand Operators (= , += , -= , *= , /= , %= , &= , |= , ^= , ~= , <<= , >>=, ++, --)
  - Relational Operators (< , <= , > , >= , !=)
  - Logical Operators (&&, ||, !)
  - Conditional Operator (? :)
  - Bitwise Operators (& , | , ^ , ~ , << , >>)
  - Special Operator (, , sizeof(), [ ], * , & , →)

- Types of operators
  - Unary Operators (+ , - , ++, --, &, *)
  - Binary Operators (+, -, *, +=, …)
  - Ternary Operators (? :)

Operators Precedence and Associativity



OPERATOR	TYPE	ASSOCIAVITY
( ) [ ] . ->		left-to-right
++ -- + - ! ~ (*type*) * & sizeof	Unary Operator	right-to-left
* / %	Arithmetic Operator	left-to-right
+ -	Arithmetic Operator	left-to-right
<< >>	Shift Operator	left-to-right
< <= > >=	Relational Operator	left-to-right
== !=	Relational Operator	left-to-right
&	Bitwise AND Operator	left-to-right
^	Bitwise EX-OR Operator	left-to-right
|	Bitwise OR Operator	left-to-right
&&	Logical AND Operator	left-to-right
||	Logical OR Operator	left-to-right
? :	Ternary Conditional Operator	right-to-left
= += -= *= /= %= &= ^= |= <<= >>=	Assignment Operator	right-to-left
,	Comma	left-to-right

Arithmetic operators

- Arithmetic operators work with all primitive data types i.e. int, float, char, double.
- Precedence of * & / is higher than + & -.

- % operator doesn't work with float and double type.
- % operator follows sign of numerator

- If two operands are of different types, the lower type is promoted temporarily for computation.

- char and short are promoted ~~is promoted~~ to int temporarily for computation.

- Char types are treated as integers (ASCII values) for calculation.

- If result exceed range of data type (overflow), then it rollback.

Control Statements

- Decision or Selection
    - if-else
    - switch-case
- Iteration (loop)
    - for
    - while
    - do-while
- Jump
    - break
    - continue
    - goto
    - return

if-else statement

```
if  (condition) {
            statement 1;
            statement 2;
}

if  (condition) {
            statement 1;
            statement 2;
}
else {
            statement 3;
            statement 4;
}
```

```
if  (condition)
            statement 1;

if  (condition)
            statement 1;
else
            statement 2;
```

- Condition is any expression – using relational, logical or other operators.
  - 0 – false condition
  - 1 – true condition

SUNBEAM

Thank you!