# Unit 2

# Data types and Expressions

Nguyen Thi Thu Huong - SoiCT - HUST

# Data types

- Standard : char, int, float, double. . .
- Constructed : array, string, struct, file . . .

# Standard Data Types

| Variable Type | Keyword | Range | Storage in Bytes |
|---|---|---|---|
| Character | char | -127 to 127 | 1 |
| Unsigned character | unsigned char | 0 to 255 | 1 |
| Unsigned integer | unsigned int | 0 to 65,535 | 2 |
| Short integer | short | -32,768 to 32,767 | 2 |
| Unsigned short integer | unsigned short | 0 to 65,535 | 2 |
| Long integer | long | -2,147,483,648 to 2,147,483,647 | 4 |
| Unsigned long | integer unsigned long | 0 to 4,294,967,295 | 4 |
| Single precision floating point | float | 1.2E-38 to 3.4E38, approx. range precision = 7 digits. | 4 |
| Double precision floating point | double | 2.2E-308 to 1.8E308, approx. range precision = 19 digits. | 8 |

# CONSTANTS

A convenient way to associate constant values with names is using the #define statement.

#define

Examples

#define TRUE 1

#define TABLESIZE 100

# Operators

- Arithmetic
- Assignment
- Logical/relational
- Bitwise

# ASSIGNMENT OPERATORS

| | |
|---|---|
| = | Assignement |
| *= | Multiply |
| /= | Divide. |
| %= | Modulus. |
| += | Add. |
| -= | Subtract. |
| <<= | Left shift. |
| >>= | Right shift. |
| &= | Bitwise AND. |
| ^= | Bitwise exclusive OR (XOR). |
| \|= | Bitwise inclusive OR. |

# Logical and Relational Operators

| | |
|---|---|
| = = | Equal to |
| != | Not equal to |
| > | Greater than |
| < | Less than |
| >= | Greater than or equal to |
| <= | Less than or equal to |
| && | Logical AND |
| \|\| | Logical OR |
| ! | Logical NOT |

# Assignment Operators

| | |
|---|---|
| = | Assignment |
| *= | Multiply |
| /= | Divide. |
| %= | Modulus. |
| += | Add. |
| -= | Subtract. |
| <<= | Left shift. |
| >>= | Right shift. |
| &= | Bitwise AND. |
| ^= | Bitwise exclusive OR (XOR). |
| \|= | Bitwise inclusive OR. |

# Logical and Relational Operators

**= =** Equal to
**!=** Not equal to
**>** Greater than
**<** Less than
**>=** Greater than or equal to
**<=** Less than or equal to
**&&** Logical AND
**||** Logical OR
**!** Logical NOT

# BITWISE OPERATORS

| | |
|---|---|
| & | AND (Binary operator) |
| \| | inclusive OR |
| ^ | exclusive OR |
| << | shift left. |
| >> | shift right. |
| ~ | one's complement |

# Increment and Decrement Operators

- The increment operator ++ adds 1 to its operand
- The decrement operator - - subtract 1 from its operand
- ++ may be used either as prefix operators or postfix operators
  - the effect is to increment n.
  - ++n increments n before its value is used,
  - n++ increment n after its value has been used.
- We also have prefix and postfix operators for - -

# Precedence of Operators

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| () | [] | -> | . | ++ | -- (postfix) | | | | |
| ! | ~ | ++ | -- (prefix) | * | & | sizeof | | | |
| * | / | % | | | | | | | |
| + | - | | | | | | | | |
| << | >> | | | | | | | | |
| < | <= | >= | > | == | != | | | | |
| & | | | | | | | | | |
| ^ | | | | | | | | | |
| \| | | | | | | | | | |
| && | | | | | | | | | |
| \|\| | | | | | | | | | |
| ?: | | | | | | | | | |
| = | += | -= | *= | /= | %= | &= | ^= | \|= | <<= |
| | | >>= | | | | | | | |

# TYPE CONVERSIONS

- If either operand is long double, convert the other to long double.
- Otherwise, if either operand is double, convert the other to double.
- Otherwise if either operand is float, convert the other to float.
- Otherwise convert char and short to int.
- Then if either operand is long, convert the other to long.
- A char is just a small integer, so chars may be freely used in arithmetic expressions.