

w03-Lec

Variables, Data Types, Expression, and Assignment Part I

Assembled for 204111
by Areerat Trongratsameethong

Ref: Introduction to Computation and Programming Using Python, revised and expanded edition By John V Guttag

Syntax vs Semantics (2)

- **Syntax**
 - ในทางภาษา
 - ไวยากรณ์:
 - เช่น หน้าจันตีม << ไม่เป็นประโยค (ต้องวาง subject verb object)
 - ในทาง programming
 - `4 5 == 2` // incorrect syntax
 - `int y = 4 * 5;` // correct syntax
- **Static Semantics**
 - บอกว่าประโยคที่ถูกต้องตามหลัก **grammar** หรือ **syntax** นั้น มีความหมายหรือไม่
 - ในทางภาษา
 - เช่น ฝนดาวปลุกอากาศ // อะไรคือความหมาย??????
 - ในทาง programming
 - `4 / "abc"` // Syntax ถูกต้อง <literal> <operator> <literal>
// แต่ไม่สื่อความหมาย

Ref: Introduction to Computation and Programming Using Python, revised and expanded edition By John V Guttag

Syntax vs Semantics

- Each programming language has a set of primitive constructs, a syntax, a static semantics, and semantics.
- เราจะอธิบาย concept โดยเปรียบเทียบกับภาษาที่ใช้ในชีวิตประจำวัน
- **Primitive constructs**
 - ในทางภาษา
 - Set of words: cat dog คน น้ำชา รถยนต์
 - ในทาง programming
 - `3.0 x y + ==` operators, operands, keywords... etc

Ref: Introduction to Computation and Programming Using Python, revised and expanded edition By John V Guttag

Syntax vs Semantics (3)

- **Semantics หรือ Full Semantics**
 - ความหมายของ **expression** หรือ ประโยคที่ถูก **syntax** และ **static semantics**
 - ในทางภาษา
 - ข้าวเย็นหมดแล้ว // ความหมายกำกวม
 - ในทาง Programming
 - อะไรคือความหมายของ **expression** นั้น ๆ
// ถูก syntax และถูก static semantics
 - `bmi = weight / height * height`
// แต่ให้ผลคำนวณที่ผิด
// ควรเป็น `bmi = weight / (height * height)`

Syntax vs Semantics (4)

- Although **syntax errors** are the most common kind
 - They are the least dangerous kind and easiest to catch
- We get some help in catching static semantic errors from compiler/interpreter
- Most of the time we have to **catch semantic errors ourselves**

5

Back to C programming

7

Summary

- **Syntax**
 - ถูกหลักไวยากรณ์ไหม?
- **Static Semantics**
 - สื่อความหมายไหม?
- **Semantics (หรือ Full Semantics)**
 - ความหมายที่สื่อคืออะไร?
 - (ตรงตามที่ต้องการไหม?)

6

Identifiers

- การกำหนดชื่อในส่วนต่าง ๆ ของโปรแกรม เช่น ชื่อโปรแกรม, ชื่อฟังก์ชัน, และชื่อตัวแปร เป็นต้น
- **Identifiers in C consist of three types:**
 - **Reserved words**
 - **Standard identifiers**
 - **Programmer-created identifiers**

8

Identifiers (2)

- **Reserved words:**
 - ส่วนของคำสั่งต่าง ๆ ที่กำหนดขึ้นโดยภาษา C ประกอบด้วย การกำหนดชนิดข้อมูล หรือ คำสั่งต่าง ๆ เช่น int, float, if เป็นต้น
- **Standard identifiers:**
 - ส่วนของชื่อฟังก์ชันต่าง ๆ ที่กำหนดขึ้นโดยภาษา C ซึ่งจัดเก็บอยู่ใน Standard Library เช่น printf, scanf เป็นต้น
- **Programmer-created identifiers:**
 - ชื่อในส่วนต่าง ๆ ของโปรแกรม เช่น ชื่อโปรแกรม ชื่อฟังก์ชัน ชื่อตัวแปร ที่กำหนดโดยผู้พัฒนาโปรแกรม

C Reserved Words

auto	const	double	float	int	short	struct
break	continue	else	for	long	signed	switch
case	default	enum	goto	register	sizeof	typedef
char	do	extern	if	return	static	union
unsigned	void	volatile	while			

Reserved Words

- **Reserved word:** word that is **predefined by the programming language** for a special purpose and can only be used in a specified manner for its intended purpose
- Also referred to as **keywords in C**
- ใช้สำหรับกำหนดชนิดข้อมูล และคำสั่งต่าง ๆ ที่สงวนไว้ ใช้สำหรับวัตถุประสงค์ที่เฉพาะเจาะจง ประกาศโดยภาษาคอมพิวเตอร์
- ผู้พัฒนาโปรแกรม **ห้าม** นำ Reserved Word หรือ Keyword ไปใช้ตั้งชื่อตัวแปรของตนเอง

Standard Identifiers

- **Standard identifiers:** words **predefined in C**
- Most of the standard identifiers are the names of functions that are provided in the **C standard library**
- It is good programming practice to use standard identifiers only for their intended purpose
- **Standard identifiers:** ส่วนใหญ่คือ ชื่อฟังก์ชันที่ประกาศไว้ใน Standard Library ของภาษา C
- ผู้พัฒนาโปรแกรม **ห้าม** นำ ชื่อ function ที่กำหนดไว้ใน C Standard Library ไปใช้ตั้งชื่อส่วนต่าง ๆ ในโปรแกรมของตนเอง

Example of C Standard Identifier

abs	fclose	gets	memcpy	rewind	strcpy
argc	fopen	isacii	printf	scanf	strlen
argv	free	isalpah	puts	sin	tolower
calloc	fseek	malloc	rand	strcat	toupper

C's identifier rules

- Identifier can be any combination of letters, digits, or underscores (`_`) subject to the following rules:
 - First character must be a letter (a-zA-Z) or underscore (`_`)
 - Only letters (a-zA-Z), digits (0-9), or underscores(`_`) may follow the initial character
 - Blank spaces are not allowed
 - Cannot be a reserved word

Programmer-created Identifiers

- Programmer-created identifiers: selected by the programmer
 - Also called **programmer-created names**
 - Used for naming data and functions
 - Must conform to **C's identifier rules**
- Programmer-created identifiers: ชื่อต่าง ๆ ที่ผู้พัฒนาโปรแกรมเป็นผู้กำหนดเอง เช่น ชื่อตัวแปร และชื่อ function ที่ผู้พัฒนาโปรแกรมสร้างเอง เป็นต้น ต้องกำหนดให้สอดคล้องกับกฎการตั้งชื่อของภาษา C

C's identifier rules (2)

- All uppercase letters usually (but not limited to) indicate a constant.
 - `#define PI 3.1412;`
 - `const int FAC_OF_SCI_ID = 05;`
 - `const int DAYS_IN_WEEK = 7;`
- A function name must be followed by parentheses (`()`)
- An identifier **should be descriptive**:
 - `degToRadians();` ← GOOD
 - `easy, duh, justDoIt` ← NOT GOOD
- C is a **case-sensitive** language
 - `TOTAL` and `total` represent different identifiers.

Example of Programmer-created Identifiers

- Valid C Identifier (ถูกต้อง)
 - calculateArea();
 - height_of_rectangle
 - BMI
 - _log
- invalid C Identifier (ไม่ถูกต้อง)
 - 4ab7
 - calculate total
 - while
 - height-of-rectangle
 - scanf
- Valid C Identifier but not informative (ถูกต้องแต่ไม่สื่อความหมาย)
 - Jameji
 - aaaa
 - xyx

Data Type

ชนิดข้อมูล

- Set of values and a set of operations that can be applied to these values
 - ชนิดของข้อมูล และการดำเนินการประเภทใดบ้างที่กระทำกับข้อมูลในแต่ละชนิดได้
- Built-in data type: is provided as an integral part of the language; also known as **Primitive Type**
 - ชนิดของข้อมูลที่เป็นส่วนหนึ่งของภาษา เราเรียกชนิดข้อมูลเหล่านี้ว่า Primitive Type

Program 2.1

```

1 #include <stdio.h>
2
3 int main()
4 {
5     /* declare an input and output item */
6     float radius, circumference;
7     /* assign value to radius */
8     radius = 2.0;
9     /* calculate the circumference */
10    circumference = 2.0 * 3.1416 * radius;
11    /* display circumference */
12    printf("The circumference of the circle is %f\n", circumference);
13
14    return 0;
15 }

```

Reserved words → ?

Standard identifiers → ?

Programmer-created identifiers → ?

Basic Data Types in C

char	a single byte, capable of holding one character in the local character set
int	an integer, typically reflecting the natural size of integers on the host machine
float	single-precision floating point
double	double-precision floating point

Qualifiers in Basic Data Types

- **Qualifiers:** เราสามารถกำหนดคุณสมบัติเพิ่มเติม (qualifier) ให้กับข้อมูลที่เป็น basic types ได้
- **Qualifiers in C**
 - short } ใช้กับ int, float, long, double (long only)
 - long }
 - signed } ใช้กับ all numerical data types
 - unsigned }

C Qualifiers: signed and unsigned

Qualifiers	Data Types	Meaning
signed	short, int, long, long long, float, double, long double	negative or positive or zero number
	char	have values between -128 and 127
unsigned	short, int, long, long long, float, double, long double	positive or zero number
	char	have values between 0 and 255

C Qualifiers: short and long

- **Provide different lengths of integers**
 - Applied to **int** datatype
 - size of **short int** \leq **int** \leq **long int**
 - Applied to **float** and **double** datatype
 - size of **float** \leq **double** \leq **long double**

Numerical Data Types

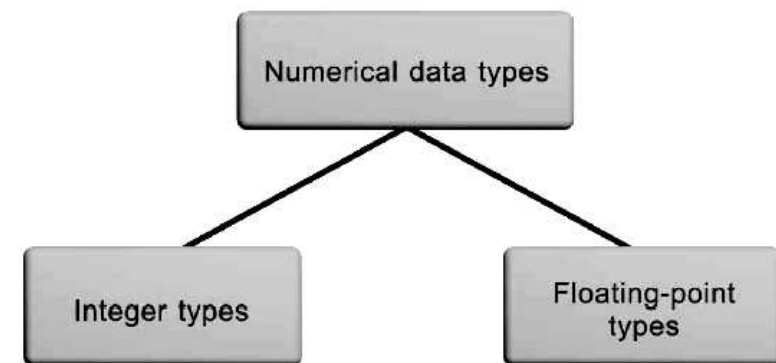


Figure 2.6 Built-in data types

- **Integer \neq int**
 - Integer (จำนวนเต็ม) สามารถมี type อื่นได้นอกจาก int
- **Floating Point \neq float**
 - Floating-point (จำนวนจริง) สามารถมี type อื่นได้นอกจาก float

Integral Data Types

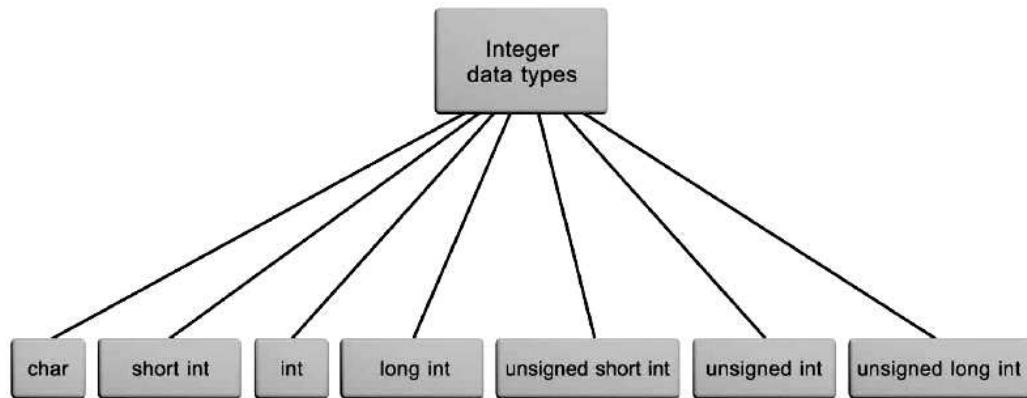


Figure 2.7 C's integer data types

Integral Data Types: char

- **char: stores individual characters (ASCII)**

- For example: 'A', '\$', 'b', '!'

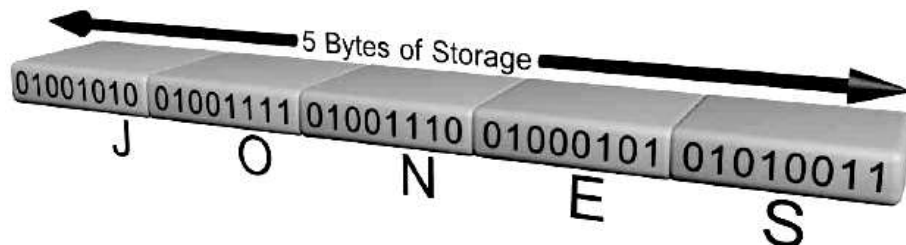
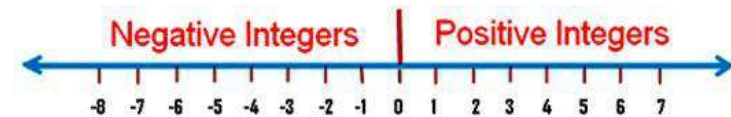


Figure 2.8 The letters JONES stored inside a computer

Integral Data Types: int

- **int: whole numbers (integers - เลขจำนวนเต็ม)**
- For example: 0, -10, 253, -26351
- Not allowed: commas, decimal points, special symbols



Integral Data Types: char(2)

Table 2.4 ASCII and ANSI Letter Codes

Letter	Code	Letter	Code	Letter	Code	Letter	Code
a	01100001	n	01101110	A	01000001	N	01001110
b	01100010	o	01101111	B	01000010	O	01001111
c	01100011	p	01110000	C	01000011	P	01010000
d	01100100	q	01110001	D	01000100	Q	01010001
e	01100101	r	01110010	E	01000101	R	01010010
f	01100110	s	01110011	F	01000110	S	01010011
g	01100111	t	01110100	G	01000111	T	01010100
h	01101000	u	01110101	H	01001000	U	01010101
i	01101001	v	01110110	I	01001001	V	01010110
j	01101010	w	01110111	J	01001010	W	01010111
k	01101011	x	01111000	K	01001011	X	01011000
l	01101100	y	01111001	L	01001100	Y	01011001
m	01101101	z	01111010	M	01001101	Z	01011010

Escape sequences (\)

Escape sequence are used to represent certain special characters

Table 2.5 Escape Sequences

Escape Sequence	Character Represented	Meaning	ASCII Code
<code>\n</code>	Newline	Move to a new line	00001010
<code>\t</code>	Horizontal tab	Move to next horizontal tab setting	00001001
<code>\v</code>	Vertical tab	Move to next vertical tab setting	00001011
<code>\b</code>	Backspace	Move back one space	00001000
<code>\r</code>	Carriage return	Carriage return (moves the cursor to the start of the current line—used for overprinting)	00001101
<code>\f</code>	Form feed	Issue a form feed	00001100
<code>\a</code>	Alert	Issue an alert (usually a bell sound)	00000111
<code>\\</code>	Backslash	Insert a backslash character (places an actual backslash character within a string)	01011100
<code>\?</code>	Question mark	Insert a question mark character	00111111
<code>\'</code>	Single quotation	Insert a single quote character (places an inner single quote within a set of outer single quotes)	00100111
<code>\"</code>	Double quotation mark	Insert a double quote character (places an inner double quote within a set of outer double quotes)	00100010
<code>\nnn</code>	Octal number	The number <i>nnn</i> (<i>n</i> is a digit) is to be considered an octal number	—
<code>\xhhh</code>	Hexadecimal number	The number <i>hhh</i> (<i>h</i> is a digit) is to be considered a hexadecimal number	—
<code>\0</code>	Null character	Insert the null character, which is defined as having the value 0	00000000

A First Book of ANSI C, 4th Edition

29

Integral Data Type Ranges

Data Type	size (byte)	Minimum	Maximum
<code>short int</code>	2	-32,768	32,767
<code>unsigned short int</code>	2	0	65,535
<code>unsigned int</code>	4	0	4,294,967,295
<code>int</code>	4	-2,147,483,648	2,147,483,647
<code>long int</code>	4	-2,147,483,648	2,147,483,647
<code>signed char</code>	1	-128	127
<code>unsigned char</code>	1	0	255

A First Book of ANSI C, 4th Edition

30

Floating-Point Data Types

- A floating-point value (real number - จำนวนจริง) can be the number zero or any positive or negative number that contains a decimal point
 - For example: +10.625, 5., -6.2, 3251.92, +2
 - Not allowed: commas, decimal points, special symbols

A First Book of ANSI C, 4th Edition

31

Floating-Point Data Types (2)

- `float`: single-precision number
- `double`: double-precision number
- Storage allocation for each data type depends on the compiler (use `sizeof()`)

Data Types	Size (bytes)
<code>float</code>	4
<code>double</code>	8
<code>long double</code>	12

represent mantissa and exponent

$$1.2345 = \underbrace{12345}_{\text{mantissa}} \times 10^{-4}$$

exponent

A First Book of ANSI C, 4th Edition

32

Floating-Point Data Types (3)

- float literal (ค่าคงที่ เช่น 2.3) is indicated by appending an f or F
- long double is created by appending an l or L
 - 9.234 indicates a double literal
 - 9.234f indicates a float literal
 - 9.234L indicates a long double literal
- สังเกตว่า floating-point literal ถ้าไม่มี suffix ใด ๆ จะมีชนิดเป็น **double**
- แต่กรณี integer literal ถ้าไม่มี suffix จะมีชนิดเป็น **int**

Floating-Point Data Type Ranges

- การกำหนดค่า floating point เป็นไปตามมาตรฐาน IEEE 754 ปี 1985

level	width	range at full precision	precision*
single precision	32 bits	$\pm 1.18 \times 10^{-38}$ to $\pm 3.4 \times 10^{38}$	approx. 7 decimal digits
double precision	64 bits	$\pm 2.23 \times 10^{-308}$ to $\pm 1.80 \times 10^{308}$	approx. 15 decimal digits

Floating-Point Data Types (4)

- Exponential Notation
 - In numerical theory, the term precision typically refers to numerical accuracy

Decimal Notation	Exponential Notation
1625.	1.625e3
63421.	6.3421e4
.00731	7.31e-3
.000625	6.25e-4

Variables and Declarations

- **Variables** are names given by programmers to computer storage
- Variable name usually **limited to 255 characters**
- Variable names are **case sensitive**

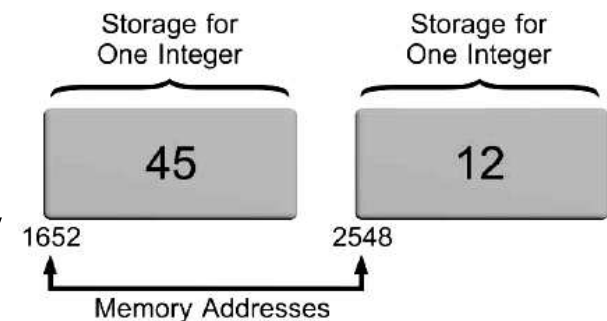


Figure 2.9 Enough storage for two integers

Declaration Statements

- Naming and specifying the data type that can be stored in each variable is accomplished using declaration statements
- การประกาศตัวแปร ประกอบด้วย การกำหนดชนิดของตัวแปร และชื่อตัวแปร เช่น

```
int num;
float height;
long double weight;
```

```
function name()
```

```
{
    declaration statements;
    other statements;
}
```

Definition statements define or tell the compiler how much memory is needed for data storage

Declaration Statements (3)

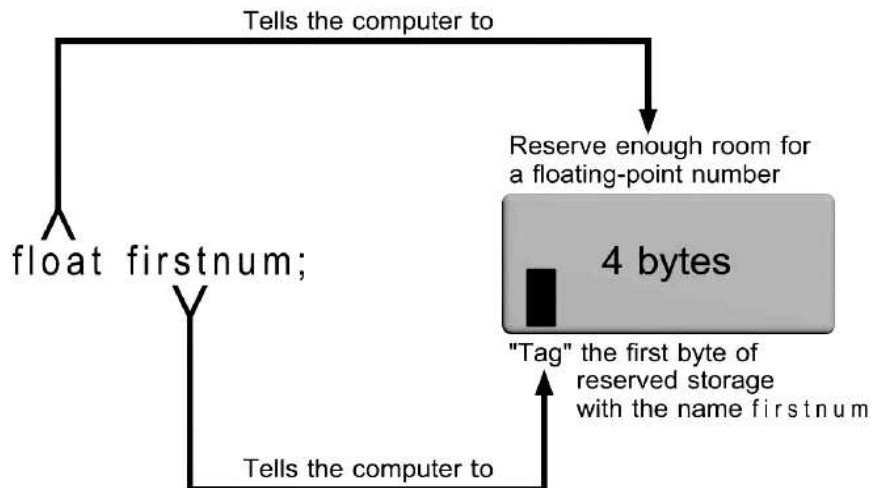


Figure 2.11b Defining the floating-point variable named `firstnum`

Declaration Statements (2)

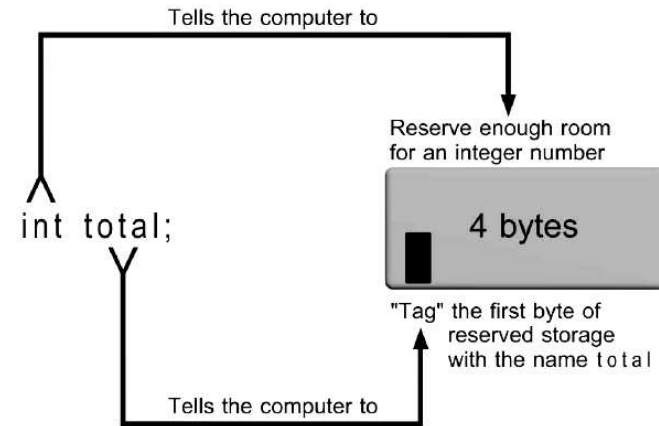


Figure 2.11a Defining the integer variable named `total`

Declaration Statements (4)

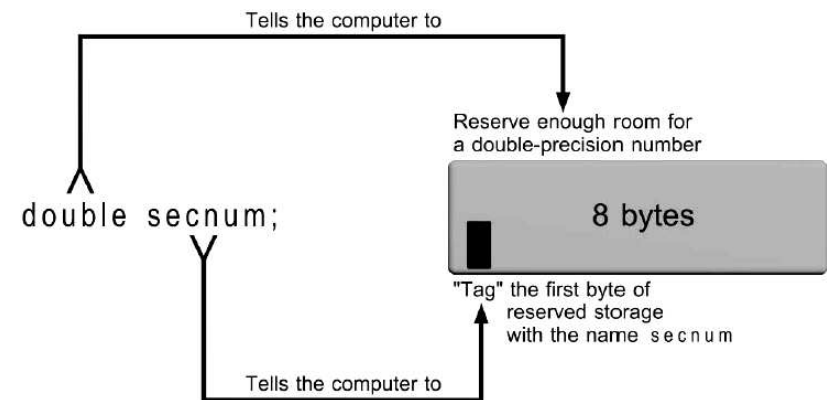


Figure 2.11c Defining the double-precision variable named `secnum`

Assignment Statement

- **Assignment Statement:** คำสั่งที่ใช้สำหรับกำหนดค่าข้อมูลให้กับตัวแปร
- `num1 = 45;`
- `num2 = 12;`
- `total = num1 + num2;`

} Assignment Statements

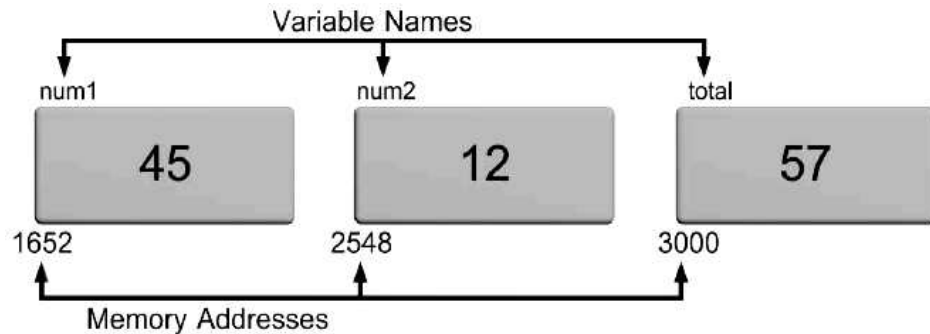


Figure 2.10 Naming storage locations

45

Variables and Assignment



Program 2.7

```

1 #include <stdio.h>
2 int main()
3 {
4     float grade1; /* declare grade1 as a double variable */
5     float grade2; /* declare grade2 as a double variable */
6     float total; /* declare total as a double variable */
7     float average; /* declare average as a double variable */
8
9     grade1 = 85.5f;
10    grade2 = 97.0f;
11    total = grade1 + grade2;
12    average = total/2.0;
13    printf("The average grade is %f\n",average);
14
15    return 0;
16 }

```

You can omit the `f` and let the compiler convert the double precision value into a float value when the assignment is made

A First Book of ANSI C, Fourth Edition

46

Arithmetic Operations

- **Arithmetic operators:** operators used for arithmetic operations:
 - Addition +
 - Subtraction -
 - Multiplication *
 - Division /
 - Modulo %
- Binary operators require two operands
- An operand can be either a literal value or an identifier that has a value associated with it

47

Arithmetic Operations (2)

- A simple binary arithmetic expression consists of a binary arithmetic operator connecting two literal values in the form:
 - `literalValue operator literalValue`
 - `3 + 7`
 - `12.62 - 9.8`
 - `.08 * 12.2`
 - `12.6 / 2.0`
- Spaces around arithmetic operators are inserted for clarity and can be omitted without affecting the value of the expression
 - `3+5` or `3 + 5` ให้ความหมายเหมือนกันและสามารถใช้ได้ทั้งคู่

48

Arithmetic Operations (3)

- Operation สำหรับข้อมูลชนิดตัวเลข

Table 2.3 C's Built-in Data Types

Data Type	Supplied Operations
Integer	+, -, *, /, %, =, ==, !=, <=, >=, sizeof(), and bit operations (see Sec. 14.2)
Floating Point	+, +, -, *, /, =, ==, !=, <=, >=, sizeof()

Expression Types

- Expression: any combination of operators and operands that can be evaluated to yield a value
- **Integer expression**: contains only integer operands; the result is an integer
- **Floating-point expression**: contains only floating-point operands; the result is a double-precision

Expression Types (2)

In a mixed-mode expression (expression ที่มี data type มากกว่า 1 ชนิด) the data type of each operation is determined by the following rules:

- 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.

Expression Types (3)

- Otherwise, convert char and short to int.
- Then, if either operand is long, convert the other to long.

สรุป

long double > double > float > long > int > (char, short)

- Lower type ถูก promote ไปเป็น higher type เสมอ
- กรณี char และ short (char vs char, char vs short, short vs short) จะได้ผลลัพธ์เป็น int
- กรณี unsigned จะมี กฎเพิ่มเติมจากนี้

Integer Division and Modulo

- $15/2 = 7$
 - Integers cannot contain a fractional part
 - Remainder is truncated (ตัดเศษทิ้งเสมอ) เช่น
 - $1/2$ // มีค่า 0
 - $59/10$ // มีค่า 5
 - $-15/2$ // มีค่า -7
- % is the modulo or remainder operator
 - $9 \% 4$ is 1
 - $17 \% 3$ is 2
 - $14 \% 2$ is 0

Arithmetic Operator Summary

Table 2.9 Summary of Arithmetic Operators

Operation	Operator	Type	Operand	Result
Addition	+	Binary	Both are integers One operand is not an integer	Integer Double-precision
Subtraction	-	Binary	Both are integers One operand is not an integer	Integer Double-precision
Multiplication	*	Binary	Both are integers One operand is not an integer	Integer Double-precision
Division	/	Binary	Both are integers One operand is not an integer	Integer Double-precision
Modulus	%	Binary	Both are integers One operand is not an integer	Integer Double-precision
Negation	B	Unary	Integer or floating point	Same as operand

Negation

- A unary operator is one that operates on a single operand, e.g., negation (-)
- The minus sign in front of a single numerical value negates (reverses the sign of) the number