Lab1: A Simple Program

Fall 2020
Sept. 15, 2020

International Bachelor Program in Informatics
College of Informatics
Yuan Ze University

Rung-Bin Lin

Review of Chapter 2

What you Learn in Chapter 2

A Program you Learned

```
1  // Fig. 2.5: fig02_05.cpp
2  // Addition program that displays the sum of two integers.
3  #include <iostream> // allows program to perform input and output
4
5  // function main begins program execution
6  int main()
7  {
8     // variable declarations
9     int number1; // first integer to add
10    int number2; // second integer to add
11    int sum; // sum of number1 and number2
12
13    std::cout << "Enter first integer: "; // prompt user for data
14    std::cin >> number1; // read first integer from user into number1
15
16    std::cout << "Enter second integer: "; // prompt user for data
17    std::cin >> number2; // read second integer from user into number2
18
19    sum = number1 + number2; // add the numbers; store result in sum
20
21    std::cout << "Sum is " << sum << std::endl; // display sum; end line
22 } // end function main
```

Fig. 2.5 | Addition program that displays the sum of two integers. (Part 1 of 2.)

2.5 Memory Concepts

- Variable names such as `number1`, `number2` and `sum` actually correspond to locations in the computer's memory.
- Every variable has a name, a type, a size and a value.
- When a value is placed in a memory location, the value overwrites the previous value in that location; thus, placing a new value into a memory location is said to be destructive.
- When a value is read out of a memory location, the process is nondestructive.

Concept of Memory

Fig. 2.6 | Memory location showing the name and value of variable `number1`.

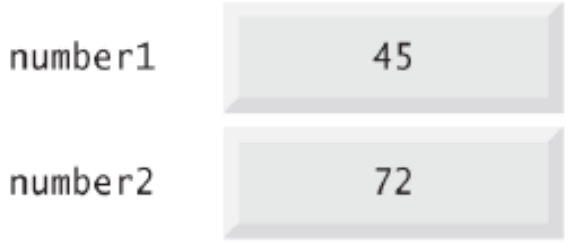

Fig. 2.7 | Memory locations after storing values for `number1` and `number2`.

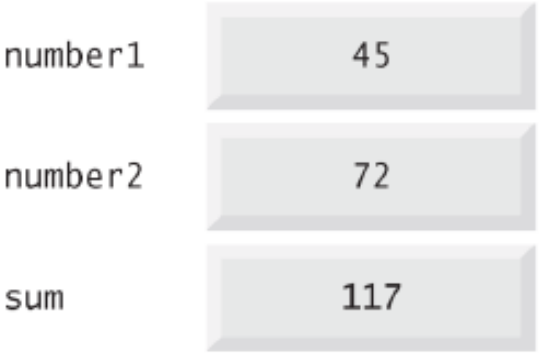

Fig. 2.8 | Memory locations after calculating and storing the `sum` of `number1` and `number2`.

Arithmetic Operators

C++ operation	C++ arithmetic operator	Algebraic expression	C++ expression
Addition	+	$f + 7$	`f + 7`
Subtraction	-	$p - c$	`p - c`
Multiplication	*	bm or $b \cdot m$	`b * m`
Division	/	x / y or $\frac{x}{y}$ or $x \div y$	`x / y`
Modulus	%	$r \bmod s$	`r % s`

Fig. 2.9 | Arithmetic operators.

Precedence of Arith. Operators

Operator(s)	Operation(s)	Order of evaluation (precedence)
( )	Parentheses	Evaluated first. If the parentheses are nested, the expression in the innermost pair is evaluated first. If there are several pairs of parentheses "on the same level" (i.e., not nested), they're evaluated left to right.
*, /, %	Multiplication, Division, Modulus	Evaluated second. If there are several, they're evaluated left to right.
+	Addition	Evaluated last. If there are several, they're evaluated left to right.
-	Subtraction	

Fig. 2.10 | Precedence of arithmetic operators.

2.7 Decision Making: Equality and Relational Operators

- The `if` statement allows a program to take alternative action based on whether a condition is true or false.
- If the condition is true, the statement in the body of the `if` statement is executed.
- If the condition is false, the body statement is not executed.
- Conditions in `if` statements can be formed by using the equality operators and relational operators summarized in Fig. 2.12.
- The relational operators all have the same level of precedence and associate left to right.
- The equality operators both have the same level of precedence, which is lower than that of the relational operators, and associate left to right.

Decision Statements

```
if (x) {
        y;
}
```

```
if (x) {
        y;
}
else  {
        z;
}
```

```
if (x) {
        y;
}
else if (x) {
        z;
}
```

Standard algebraic equality or relational operator	C++ equality or relational operator	Sample C++ condition	Meaning of C++ condition
Relational operators			
>	>	x > y	x is greater than y
<	<	x < y	x is less than y
≥	>=	x >= y	x is greater than or equal to y
≤	<=	x <= y	x is less than or equal to y
Equality operators			
=	==	x == y	x is equal to y
≠	!=	x != y	x is not equal to y

Fig. 2.12 | Equality and relational operators.

LAB 1: A Simple Program

LAB 1: A simple program

- **Write a C++ program to do the following computations or tasks:**
  - Get three integers from keyboard.
  - Multiply the first two integers and store the result
  - Divide the first integer by the second integer and store the result if the result is available
  - Perform a modulus operation on the second and the third integers and store the result if the result is available.
  - Add the first, the second integer and third integer and store the result if the result is available.
  - Display the sum of the above four results if all the above results are available. Otherwise, print “Some results are not available!”.

Input & Output Examples

Sample inputs	Sample outputs
0 12 7	24
12 8 0	Some results are not available!
34 87 101	3267
54 0 -23	Some results are not available!
-13 -3 7	31

Code Submission

- Submit your program to the course portal according to the submission rules. For example, name the submitted file S1093234_LAB1.zip for this lab, where 1093234 should be replaced by your student ID number.