Lab1 Introduction to C++ Programming

Objectivities:
1. To write simple computer programs in C++.
2. To write simple input and output statements.
3. To use fundamental types.
4. Basic computer memory concepts.
5. To use arithmetic operators.
6. The precedence of arithmetic operators.
7. To write simple decision-making statements

Experiment

- **Ex1:**

输入书上 p38 例 fig02_03.cpp，熟悉编程环境。

- **Ex2:**

Description of the Problem
Write a program that inputs three integers from the keyboard, and prints the sum, average, product, smallest and largest of these numbers. The screen dialogue should appear as follows: [Note: 13, 27 and 14 are input by the user.]

Sample Output
```plaintext
Input three different integers: 13 27 14
Sum is 54
Average is 18
Product is 4914
Smallest is 13
Largest is 27
```

Template
```cpp
#include <iostream> // allows program to perform input and output
using std::cout; // program uses cout
using std::endl; // program uses endl
using std::cin; // program uses cin

int main()
{
```
10 int number1; // first integer read from user
11 int number2; // second integer read from user
12 int number3; // third integer read from user
13 int smallest; // smallest integer read from user
14 int largest; // largest integer read from user
15
16 cout << "Input three different integers: "; // prompt
17 /* Write a statement to read in values for number1, number2 and
18 number3 using a single cin statement */
19
20 largest = number1; // assume first integer is largest
21
22 /* Write a statement to determine if number2 is greater than
23 largest. If so assign number2 to largest */
24
25 /* Write a statement to determine if number3 is greater than
26 largest. If so assign number3 to largest */
27
28 smallest = number1; // assume first integer is smallest
29
30 /* Write a statement to determine if number2 is less than
31 smallest. If so assign number2 to smallest */
32
33 /* Write a statement to determine if number3 is less than
34 smallest. If so assign number3 to smallest */
35
36 /* Write an output statement that prints the sum, average,
37 product, largest and smallest */
38
39 return 0; // indicate successful termination
40
41 } // end main

Problem-Solving Tips
1. Prompt the user to input three integer values. You will use a single cin statement to
read all three values.
2. Sometimes it is useful to make an assumption to help solve or simplify a problem.
   For example, we assume number1 is the largest of the three values and assign it to
   largest. You will use if statements to determine whether number2 or number3 are
   larger.
3. Using an if statement, compare largest to number2. If the content of number2 is
   larger, then store the variable’s value in largest.
4. Using an if statement, compare largest to number3. If the content of number3 is
larger, then store the variable’s value in largest. At this point you are guaranteed to have the largest value stored in largest.
5. Perform similar steps to those in Steps 2–4 to determine the smallest value.
6. Write a cout statement that outputs the sum, average, product (i.e., multiplication), largest and smallest values.
7. Be sure to follow the spacing and indentation conventions mentioned in the text.
8. If you have any questions as you proceed, ask your lab instructor for assistance.

Follow-Up Questions and Activities
1. Modify your solution to use three separate cin statements rather than one. Write a separate prompt for each cin.

2. Does it matter whether < or <= is used when making comparisons to determine the smallest integer? Which did you use and why?

● Ex3:

Description of the Problem
Write a program that reads in two integers and determines and prints whether the first is a multiple of the second. [Hint: Use the modulus operator.]

Sample Output

<table>
<thead>
<tr>
<th>Enter two integers: 22 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>22 is not a multiple of 8</td>
</tr>
</tbody>
</table>

Problem-Solving Tips
1. The input data consists of two integers, so you will need two int variables to store the input values.
2. Use cin to read the user input into the int variables.
3. Use an if statement to determine whether the first number input is a multiple of the second number input. Use the modulus operator, %. If one number divides into another evenly, the modulus operation results in 0. If the result is 0, display a message indicating that the first number is a multiple of the second number.
4. Use an if statement to determine whether the first number input is not a multiple of the second number input. If one number does not divide into another evenly, the modulus operation results in a non-zero value. If non-zero, display a message indicating that the first number is not a multiple of the second.
5. Be sure to follow the spacing and indentation conventions mentioned in the text.
6. If you have any questions as you proceed, ask your lab instructor for assistance.

● Ex4:

Problem Description
Write a program that inputs a five-digit number, separates the number into its individual digits and prints the digits separated from one another by three spaces each.
[Hint: Use integer division and the modulus operator.] For example, if the user inputs 42339, the program should print what is shown in the sample output.

**Sample Output**

4 2 3 3 9

**Template**

```cpp
1 // ex4
2 #include <iostream> // allows program to perform input and output
3 4 using std::cout; // program uses cout
5 using std::endl; // program uses endl
6 using std::cin; // program uses cin
7 8 int main()
9 {
10 int number; // integer read from user
11 12 cout << "Enter a five-digit integer: "; // prompt
13 cin >> number; // read integer from user
14 15 /* Write a statement to print the left-most digit of the 16 5-digit number */
17 /* Write a statement that changes number from 5-digits 18 to 4-digits */
19 /* Write a statement to print the left-most digit of the 20 4-digit number */
21 /* Write a statement that changes number from 4-digits 22 to 3-digits */
23 /* Write a statement to print the left-most digit of the 24 3-digit number */
25 /* Write a statement that changes number from 3-digits 26 to 2-digits */
27 /* Write a statement to print the left-most digit of the 28 2-digit number */
29 /* Write a statement that changes number from 2-digits 30 to 1-digit */
31 cout << number << endl;
32 33 return 0; // indicate successful termination
34 35 } // end main
```
Problem-Solving Tips

1. The input data consists of one integer, so you will use an int variable (number) to represent it. Note that the description indicates that one five-digit number is to be input—not five separate digits.
2. You will use a series of statements to “break down” the number into its individual digits using modulus (%) and division (/) calculations.
3. After the number has been input using cin, divide the number by 10000 to get the leftmost digit. Why does this work? In C++, dividing an integer by an integer results in an integer. For example, 42339 / 10000 evaluates to 4 because 10000 divides evenly into 42339 four times. The remainder 2339 is truncated.
4. Change the number to a 4-digit number using the modulus operator. The number modulus 10000 evaluates to the integer remainder—in this case, the right-most four digits. For example, 42339 % 10000 results in 2339. Assign the result of this modulus operation to the variable that stores the five-digit number input.
5. Repeat this pattern of division and modulus reducing the divisor by a factor of 10 each time (i.e., 1000, 100, 10). After the number is changed to a four-digit number, divide/modulus by 1000. After the number is changed to a three-digit number, divide/modulus by 100. And so on.
6. Be sure to follow the spacing and indentation conventions mentioned in the text.
7. If you have any questions as you proceed, ask your lab instructor for assistance.

Follow-Up Questions and Activities

1. What happens when the user inputs a number which has fewer than five digits? Why? What is the output when 1763 is entered?
2. The program you completed in this lab exercise inputs a number with multiple digits and separates the digits. Write the inverse program, a program which asks the user for three one-digit numbers and combines them into a single three-digit number. [Hint: Use multiplication and addition to form the three-digit number.]