

COMP231 Advanced Programming Chapter 5 Loops

Compiled By: Dr. Majdi Mafarja Fall Semester 2017/2018

Uploaded By: anonymous

STUDENTS-HUB.com

Opening Problem

Problem:

System.out.println("Welcome to Java!"); 100 times System.out.println("Welcome to Java!"); System.out.println("Welcome to Java!"); System.out.println("Welcome to Java!");

STUDENTS-HUB.com
Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All
Uploaded By: anonymous

Introducing while Loops

```
int count = 0;
while (count < 100) {
   System.out.println("Welcome to Java");
   count++;
```



while Loop Flow Chart



animation

Trace while Loop

int count = 0;

while (count < 2) {

System.out.println("Welcome to Java!")

count++;

The loop exits. Execute the next statement after the loop.



STUDENTS-HUB.com
Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All
Uploaded By: anonymous

Problem: Repeat Addition Until Correct

Recall that Listing 3.1 AdditionQuiz.java gives a program that prompts the user to enter an answer for a question on addition of two single digits. Using a loop, you can now rewrite the program to let the user enter a new answer until it is correct.

IMPORTANT NOTE: If you cannot run the buttons, see <u>www.cs.armstrong.edu/liang/javaslidenote.doc</u>.

RepeatAdditionQuiz

Run

STUDENTS-HUB.com
Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All
Uploaded By: anonymous

Ending a Loop with a Sentinel Value

Often the number of times a loop is executed is not predetermined. You may use an input value to signify the end of the loop. Such a value is known as a *sentinel value*.

Write a program that reads and calculates the sum of an unspecified number of integers. The input 0 signifies the end of the input.

STUDENTS-HUB.com
Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All
rights reserved.
Uploaded By: anonymous

SentinelValue

Run

Caution

Don't use floating-point values for equality checking in a loop control. Since floating-point values are approximations for some values, using them could result in imprecise counter values and inaccurate results. Consider the following code for computing 1 + 0.9 + 0.8 + ... + 0.1:

double item = 1; double sum = 0; while (item != 0) { // No guarantee item will be 0 sum += item; item -= 0.1; } System.out.println(sum);

STUDENTS-HUB.com
Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All
rights reserved.
Uploaded By: anonymous



STUDENTS-HUB.com
Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All
rights reserved.
Uploaded By: anonymous

for Loops



```
animation
```

Trace for Loop, cont.





Note

The <u>initial-action</u> in a <u>for</u> loop can be a list of zero or more comma-separated expressions. The <u>action-after-each-</u> <u>iteration</u> in a <u>for</u> loop can be a list of zero or more commaseparated statements. Therefore, the following two <u>for</u> loops are correct. They are rarely used in practice, however.

for (int i = 1; i < 100; System.out.println(i++));</pre>

for (int i = 0, j = 0; (i + j < 10); i++, j++) {
 // Do something</pre>



STUDENTS-HUB.com
Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All
Uploaded By: anbaymous

Note

If the <u>loop-continuation-condition</u> in a <u>for</u> loop is omitted, it is implicitly true. Thus the statement given below in (a), which is an infinite loop, is correct. Nevertheless, it is better to use the equivalent loop in (b) to avoid confusion:



Caution

Adding a semicolon at the end of the <u>for</u> clause before the loop body is a common mistake, as shown below:

> Logic Error

for (int i=0; i<10; i++);</pre>

System.out.println("i is " + i);

STUDENTS-HUB.com
Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All
Uploaded By: anbhymous

Caution, cont.

Similarly, the following loop is also wrong: int i=0; while (i < 10); Logic Error { System.out.println("i is " + i); i++;

In the case of the <u>do</u> loop, the following semicolon is needed to end the loop. int i=0; do {

```
System.out.println("i is " + i);
i++;
```

```
} while (i<10); Correct
```

STUDENTS-HUB.com
Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All
rights reserved.
Uploaded By: anonymous

Which Loop to Use?

The three forms of loop statements, <u>while</u>, <u>do-while</u>, and <u>for</u>, are expressively equivalent; that is, you can write a loop in any of these three forms. For example, a <u>while</u> loop in (a) in the following figure can always be converted into the following <u>for</u> loop in (b):



A for loop in (a) in the following figure can generally be converted into the following while loop in (b) except in certain special cases (see Review Question 3.19 for one of them):



STUDENTS-HUB.com

Recommendations

Use the one that is most intuitive and comfortable for you. In general, a for loop may be used if the number of repetitions is known, as, for example, when you need to print a message 100 times. A while loop may be used if the number of repetitions is not known, as in the case of reading the numbers until the input is 0. A do-while loop can be used to replace a while loop if the loop body has to be executed before testing the continuation condition.

Nested Loops

Problem: Write a program that uses nested for loops to print a multiplication table.



STUDENTS-HUB.com Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All Uploaded By: anonymous

Minimizing Numerical Errors

Numeric errors involving floating-point numbers are inevitable. This section discusses how to minimize such errors through an example.

Here is an example that sums a series that starts with 0.01 and ends with 1.0. The numbers in the series will increment by 0.01, as follows: 0.01 + 0.02 + 0.03 and so on.

TestSum

Run

STUDENTS-HUB.com Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All Viploaded By: antonymous

Problem:

Finding the Greatest Common Divisor

Problem: Write a program that prompts the user to enter two positive integers and finds their greatest common divisor.

Solution: Suppose you enter two integers 4 and 2, their greatest common divisor is 2. Suppose you enter two integers 16 and 24, their greatest common divisor is 8. So, how do you find the greatest common divisor? Let the two input integers be n1 and n2. You know number 1 is a common divisor, but it may not be the greatest commons divisor. So you can check whether k (for k = 2, 3, 4, and so on) is a common divisor for n1 and n2, until k is greater than n1 or n2.

GreatestCommonDivisor

Rur



Case Study: *Converting Decimals to Hexadecimals*

Hexadecimals are often used in computer systems programming (see Appendix F for an introduction to number systems). How do you convert a decimal number to a hexadecimal number? To convert a decimal number d to a hexadecimal number is to find the hexadecimal digits h_n , h_{n-1} , h_{n-2} , ..., h_2 , h_1 , and h_0 such that

$$d = h_n \times 16^n + h_{n-1} \times 16^{n-1} + h_{n-2} \times 16^{n-2} + \dots + h_2 \times 16^2 + h_1 \times 16^1 + h_0 \times 16^0$$

These hexadecimal digits can be found by successively dividing d by 16 until the quotient is 0. The remainders are h_0 , h_1 , h_2 , ..., h_{n-2} , h_{n-1} , and h_n .

Dec2Hex

Run

Using break and continue

Examples for using the break and continue keywords:

TestBreak.java



TestContinue.java

TestContinue Run



STUDENTS-HUB.com Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All Uploaded By: an20 ymous rights reserved.

break

```
public class TestBreak {
  public static void main(String[] args) {
    int sum = 0;
    int number = 0;
    while (number < 20) {
      number++;
      sum += number;
      if (sum >= 100)
       break;
    System.out.println("The number is " + number);
    System.out.println("The sum is " + sum);
```

continue

```
public class TestContinue {
  public static void main(String[] args) {
    int sum = 0;
    int number = 0;
    while (number < 20) {
      number++;
      if (number == 10 || number == 11)
      <u>    continue;</u>
     sum += number;
    System.out.println("The sum is " + sum);
```

STUDENTS-HUB.com
Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All
rights reserved.
Uploaded By: an24 hymous

Problem: Checking Palindrome

A string is a palindrome if it reads the same forward and backward. The words "mom," "dad," and "noon," for instance, are all palindromes.

The problem is to write a program that prompts the user to enter a string and reports whether the string is a palindrome. One solution is to check whether the first character in the string is the same as the last character. If so, check whether the second character is the same as the second-to-last character. This process continues until a mismatch is found or all the characters in the string are checked, except for the middle character if the string has an odd number of characters.



Problem: Displaying Prime Numbers

Problem: Write a program that displays the first 50 prime numbers in five lines, each of which contains 10 numbers. An integer greater than 1 is *prime* if its only positive divisor is 1 or itself. For example, 2, 3, 5, and 7 are prime numbers, but 4, 6, 8, and 9 are not.

Solution: The problem can be broken into the following tasks:

- •For number = 2, 3, 4, 5, 6, ..., test whether the number is prime.
- •Determine whether a given number is prime.
- •Count the prime numbers.
- •Print each prime number, and print 10 numbers per line.

PrimeNumber

Run