The *for* Loop and Practice Problems

CS 107

Stephen Majercik

**Use**

To repeat execution of a statement (possibly a compound statement) once for each value of a specified range of values. The values may or may not be used in the statement being executed. This is like a *while* loop in which all of the loop-control information (initialization-statement, repetition-condition, and update-statement) is contained in the header.

**Form**

\[
\text{for (initialization-statement ; repetition-condition ; update-statement) }
\]

\[
\text{statement;}
\]

where *for* is a reserved word, *initialization-statement* is an assignment to a variable controlling the loop iterations, *repetition-condition* is a Boolean expression specifying when iteration should take place, *update-statement* is a statement updating the value of the control variable initialized in the initialization statement, and *statement* is a Java statement (possibly a compound statement, i.e. a group of statements enclosed by curly braces).

**Action**

The *initialization statement* executes when the *for* statement begins execution. Prior to each loop repetition (including the first one), the *repetition condition* is tested. If it is *true*, the *statement*, or loop body, executes. If it is *false*, the loop is exited, and the flow of execution continues with the statement following the *for* loop. The *update statement* executes immediately after each repetition of the loop body. Note that the variable declared in the *initialization statement* is visible *only* within the loop. In other words, once you exit the loop, you cannot use this variable in any of your statements.

**Examples**

```java
// writes squares and square roots of numbers 10 down to 0
// Note that sqrt is a predefined C++ function that computes
// the square root of a number
for (int number = 10 ; number >= 0 ; --number) {
    System.out.println("Square of " + number + " = " + (number * number) + endl;
    System.out.println("Square root of " + number + " = " + Math.sqrt(number));
}
```

```java
// raise a to power b and store in exp
double exp = 1;
for (int i = 1 ; i <= b ; i++)
    exp = exp * a;
```
// print out grid of asterisks with dimensions specified by user
cout << "Enter dimensions of grid (height width): ";
int height = r.readInt();
r.readLine();
int width = r.readInt();
r.readLine();

for (int i = 1 ; i <= height ; i++) {
    for (int j = 1 ; j <= width ; j++)
        System.out.print("* ");
    System.out.println();
}

Practice Problems

• What’s wrong with the following for loop? There are at least 3 problems with the loop.

    // Supposed to add SCHRODINGERS_CONSTANT to each integer
    // from 1 to highLimit, issuing a warning when the sum
    // exceeds highLimit

    final int SCHRODINGERS_CONSTANT = 57;
    int highLimit = 20;
    for (int num == 1 ; num <= highLimit ; --num)
        System.out.println(num + " + SCHRODINGERS_CONSTANT = "+
                           num + SCHRODINGERS_CONSTANT);
        if (num + SCHRODINGERS_CONSTANT > highLimit)
            System.out.println("The sum has exceeded "+ highLimit);

• Describe the output the following for loops produce.

   (a)

        for (char ch = 'a' ; ch <= 'z' ; ch++)
            System.out.println(ch);

    Note that since chars are ordinal data types they can be incremented and decre-
    mented with the ++ and -- operators. The result of applying these operations
to a character is to change it to the next or previous character, respectively. For
every example, if the value of ch is 'd', after executing the statement ++ch; the value
of ch will be 'e'.

3
int limit = 100 / 11;
for (int num = 2; num <= limit; ++num) {
    System.out.print("Enter a multiple of " + num + ":");
    int multiple = r.readInt();
    r.readLine();
    while (multiple % num != 0) {
        System.out.print("INCORRECT! Try again: ");
        multiple = r.readInt();
        r.readLine();
    }
}

---------------------------------------------------------------------
for (char Ch = 'Z'; Ch >= 'A'; --Ch) {
    if ((Ch == 'A') || (Ch == 'E') || (Ch == 'I') ||
        (Ch == 'O') || (Ch == 'U'))
        System.out.println(Ch + " is a vowel. ");
    else
        System.out.println(Ch + " is a consonant. ");
}

---------------------------------------------------------------------
int limit = 24 % 6;
for (int control = 1; control <= limit; control++)
    System.out.println(Math.sqrt(control));

---------------------------------------------------------------------
for (int outer = 1; outer <= 5; ++outer)
    for (int inner = outer; inner <= 3; ++inner)
        System.out.println(outer + " * " + inner + " = " +
                           (outer * inner));

• Write a for loop that writes out the decimal equivalent of the reciprocals 1/2, 1/3, 1/4, ... , 1/19, 1/20.
• Write a for loop that reads in 3 non-negative integers from the user (one at a time) and assigns the maximum value to a variable maxVal.