Chapter-4

A portion of a program that repeats a statement or group of statements is called a loop. The statement or group of statements to be repeated in a loop is called the body of the loop. Each repetition of the loop body is called an iteration of the loop.

The while statement:

A while statement repeats its action again and again as long as a controlling Boolean expression is true. The loop is repeated while the controlling Boolean expression is true; When the expression is false, the repetition ends.

The statement starts with the keyword while followed by a Boolean expression in parentheses;

The body is always a compound statement enclosed in braces{};

While(Boolean\_Expression)

Body

The body of a while loop often is a compound statement;

A while loop can perform zero iterations;

The do-while statement;

Do

Body

While(Boolean\_Expression);

A do-while loop is executed, the loop body executes first;

A common program bug is a loop that does not end, but simply repeats its loop body again and again forever. A loop that iterates its body repeatedly without ever ending is called an infinite loop.

Some infinite loops will not really run forever but will instead end your program when some system resource is exhausted.

The for statement or for loop enables you to easily write a loop that is controlled by some sort of counter;

For( count = 1; count <= 3; count++)

System.out.println(count);

The first of the three expressions in parentheses , tells what happens before the loop body is executed for the first time;

The third expression, count++ is executed after each iteration of the loop body. The middle expression , count <= 3, is a Boolean expression that determines when the loop will end and it does so in the same way as the controlling Boolean expression in a while loop.

For(Initializing\_Action; Boolean\_Expression; Update\_Action)

Body

Declaring Variables within a for statement:

Int sum = 0;

For(int n = 1; n < =10; n++)

Sum = sum + n\*n ;

In this case ,the variable n is local to the for loop, meaning that it can not be used outside of the loop. Which means that you would not be able to display n in a println statement after the loop completes;

The for-each statement(future study)

Java also provides another form of the for statement for use when you have a collection of data such as an enumeration.

Enum Suit { CLUBS, DIAMONDS, HEARTS,SPADES}

For (Suit nextSuit: Suit.values())

System.out.print(nextSuit + “”);

System.out.println();

First part is the loop body:

Initializing statements

Controlling the number of loop iterations;

A user can signal the end of repetition by entering a sentinel value;

Break:

A loop can exit when it encounters a break statement. When a break statement executes, the immediately enclosing loop ends ,and the remainder of the loop body does not execute.

A continue statement within the body of a loop ends its current iteration and begins the next one. Using a continue statement in this way has the same problems as using a break statement. The continue statement, like the break statement ,can and should be avoided in a loop;

Loop bugs:

First: Unintended infinite loops;

Second: Off-by-one errors;

Off-by-one errors are caused by an incorrect Boolean expression;

An assertion is a statement that says something about the state of your program. An assertion can be either true or false but should be true if there are no mistakes in your program.

In Java ,you can acturally check to see whether an assertion is true and if it is not true, you can stop the program and display an error message. An assertion check in Java has the following form:

Assert Boolean\_Expression;

If Boolean\_Expression is true, nothing special happens and execution continues;