**Types of Loops**

lcv = loop control variable

**Sentinel Loops**

Often, it will be necessary to write loops that repeat until a certain value is entered. These are known as sentinel loops. Think of a sentinel as a red flag – “the look out man”. The loop will continue until you wave a red flag at it – you spot the value - the“enemy”. The sentinel can be a single value or it can be an entire range of values. Sentinel loops are more useful than count loops, since they allow for an undetermined amount of repetitions. There are two common places for using a sentinel loop

1. Reading an unknown amount of input from the user. This is the most flexible way to read data from the user. The program will be able to accept any amount of input from the user.
2. Validating input. If it is necessary to verify that a user's input falls within a certain range, then a sentinel loop is required. It is now known ahead of time how many times the user will enter an invalid number. It is not enough to validate the first input from the user. What if the user types another invalid input?

Sentinel loops still have three steps for the loop control variable: initialize outside the loop, test and modify within the loop to change the lcv to meet the condition.

Example in Java:

do{

sum = sum + num;

count++;

System.out.println("Enter a number, -1 will entry will calculate the average.");

num = input.nextInt();

}while(num != -1);

System.out.println("The average is " + average);

**Count Loops**

The only time to use a count loop is when the program can determine ahead of time how many times the loop will repeat. This is not always possible. There are generally two ways that the number of repetitions of a loop will be known ahead of time:

1. The loop always repeats the same number of times.
2. The program calculates the number of repetitions based upon user input.

When designing this loop, the first question to ask is, "How many times will the loop repeat?"

Format: initialize, test, modify in the for structure

Example in Java: Creating a Fibonacci series

final int limit = 20;

int f1, f2 = 0, f3 = 1;

for(int i = 1; i <= limit; i++){

System.out.print(" "+ f3 +" ");

f1 = f2;

f2 = f3;

f3 = f1 + f2;

}

## Conditional Loops

Conditional loops have common traits with sentinel and count loops. They are like sentinel loops in that it is unknown ahead of time how many times they will repeat. They are like count loops in that they terminate as the result of a calculation, instead of based upon user input.

Format: initialize outside the loop, test, modify within the loop to change the lcv to meet the condition.

Example in Java:

double invest = 100.00;

double goal = 1000000.00;

double rate = 0.06;

int years = 0;

double total = invest;

while (total < goal){

years++;

total = total \* (1 + rate);

}

System.out.println(“It will take ” + years + “ to reach the goal of ” + total);