### **Linear Search (for loop revisited)**

Assume that we have a Student class, with at least three methods:

public String getName()

public String getAddress()

Now assume that we have another class that has an array of Student objects as an instance variable.

private Student[] allStudents;

Assume also that somewhere in this unnamed class we construct and initialize this array of Student objects.  We want to implement a method called getStudentAddress that will take a student name as a parameter and return the address for that student.  We will look for the address for the given student using a "linear search."

public String getStudentAddress(String name)

{

for ( int i = 0; i < allStudents.length; i++ )

{

if ( name.equals(allStudents[i].getName()) )

return allStudents[i].getAddress();

}

// if we didn't return already, student is not in array

return null; // or return new String();

}

### **Binary Search: Example of a while loop**

If the data items we're searching through are ordered, and if they are in a random-access data structure such as an array or an ArrayList, then we can use a much faster search algorithm called binary search.  We could write a binary search to find a student's address given their name, but the syntax will be a little simpler if our example searches based on an int value rather than a String.  So let's assume that our Student class has an additional method, getID, that returns the student ID as an int.  Now we will write our search to find a student's address given their ID number.  We will also assume that the students in the allStudents array are sorted by their student ID numbers, not by their names.

public String getStudentAddress(int studentID)

{

int low = 0; // index of first element

int high = allStudents.length - 1; // index of last element

while ( low <= high )

{

int mid = (low + high) / 2;

int idOfStudentInMiddle = allStudents[mid].getID();

if ( studentID == idOfStudentInMiddle )

return allStudents[mid].getAddress();

else if ( studentID < idOfStudentInMiddle )

{

// student must be in lower half of array

high = mid - 1;

}

else

{

// student must be in upper half of array

low = mid + 1;

}

}

// if we didn't return already, student is not in array

return null; // or return new String();

}