

Keeping a running total

• Output:

How many days did you ride you bike? 3 Enter the miles for day 1: 14.2 Enter the miles for day 2: 25.4 Enter the miles for day 3: 12.2 Total miles ridden: 51.8

• How would you calculate the average mileage?

5

5.8 Sentinel controlled loop

- <u>sentinel</u>: special value in a list of values that indicates the end of the data
- sentinel value must not be a valid value! -99 for a test score, -1 for miles ridden
- User does not need to count how many values will be entered
- Requires a "priming read" before the loop starts

6

- so the sentinel is NOT included in the sum
- the loop can be skipped (if first value is the sentinel)

Sentinel example

• Example:

		l = 0.0; //Accumulator s; //daily miles ridden				
	<pre>cout << "Enter the miles you rode on your bike each day, "; cout << "then enter -1 when finished. " << endl;</pre>					
	cin >> mil					
	<pre>while (miles != -1) { total = total + miles; //skipped when miles==-1 cin >> miles; //get the next one }</pre>					
	<pre>cout << "Total miles ridden: " << total << endl;</pre>					
		Enter the miles you rode on your bike each d then enter -1 when finished. 14.2 25.4 12.2	ay,			
		-1 Total miles ridden: 51.8				

5.9 Which Loop to use?

- Any loop can work for any given problem
- while loop:
 - test at start of loop
 - validating input, sentinel controlled loops, etc.
- for loop:
 - initialize/test/update
 - count-controlled loops
- · do-while loop
 - always do at least once
 - good for repeating, simple menu processing

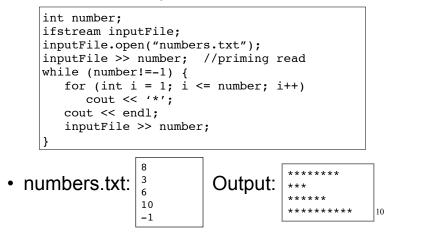
5.10 Nested loops

- When one loop appears in the body of another
- For every iteration of the outer loop, we do all the iterations of the inner loop
- Example from "real life":
- A clock. For each hour in a day (24), we iterate over 60 minutes.

12:00	1:00	2:00	3:00
12:01	1:01	2:01	
12:02	1:02	2:02	
•••	•••	•••	•
12:59	1:59	2:59	•

Print a bar graph

• Input numbers from a file. For each number, output that many asterisks (*) in a row.



Calculate grades for a class

9

For each student, input the test scores from the user and output the average.

```
int numStudents, numTests;
cout << "How many students? ";</pre>
cin >> numStudents;
cout << "How many test scores? ";</pre>
cin >> numTests;
for (int student=1; student <= numStudents; student++) {</pre>
   float total = 0, score;
   cout << "Enter the " << numTests</pre>
        << " test scores for student " << student << endl;
   for (int test=1; test <= numTests; test++) {</pre>
      cin >> score:
      total = total + score;
                                            Inner loop
   float avgScore = total/numTests;
   cout << "Average for student" << student</pre>
                                                       Outer loop
        << " is: " << avgScore << endl;
```

Calculate grades for a class

• Output:

How many students? 3 How many test scores? 4 Enter the 4 test scores for student 1 88 90.5 92 77.5 Average for student1 is: 87.0 Enter the 4 test scores for student 2 66.5 70.5 80 86 Average for student2 is: 75.8 Enter the 4 test scores for student 3 99 93.5 80 79 Average for student3 is: 87.9

5.11 More File I/O Reading data from a file Can test a file stream object as if it were a • Use fin>>x; in a loop boolean variable to check for various errors. Problem: when to stop the loop? • After opening a file, if the open operation failed, First entry in file could be count of number of the value of file stream variable is false. items problems: maintenance, large files ifstream infile: Could use sentinel value infile.open("test.txt"); problem: may not be one, maintenance if (!infile) { · Want to automatically detect end of file cout << "File open failure!";</pre> return 1; 13 14 Using >> to detect end of file Using the result of >> stream extraction operation (>>) returns true • Example: when a value was successfully read, false int number: ifstream inputFile; otherwise inputFile.open("numbers.txt"); int number: bool foundValue = (inputFile >> number); ifstream inputFile; inputFile.open("numbers.txt"); if (foundValue) bool foundValue = (inputFile >> number); cout << "The data read in was: " << number << endl;</pre> else inputFile >> number: cout << "Could not read data from file." << endl;</pre> tries to read a value into number Can also use directly as relational expression: if it was successful, value is true if (inputFile >> number) if it failed (nothing left to input), value is false (and the value in the variable does not change!) 15 16

Sum all the values in the file

Code:

```
int number;
ifstream inputFile;
inputFile.open("numbers.txt");
int total = 0;
while (inputFile >> number) {
   total = total + number;
}
cout << "The sum of the numbers in the file: " << total
   << endl;</pre>
```

```
• numbers.txt:
```

84

32 99

77

52

Output:

The sum of the numbers in the file: 344

17

Stopping a single iteration

- Sometimes we want to abort an iteration (skip to the end of loop body) before it is done.
- The continue statement can be used to terminate the current iteration:

```
for (int i=1; i <= 6; i++) {
    if (i == 4)
        continue;
    cout << i << " ";
}</pre>
```

- Output: 1 2 3 5 6
- Don't do this either. It makes your code hard to read and debug.

5.12 Breaking and Continuing

- Sometimes we want to abort (exit) a loop before it has completed.
- The break statement can be used to terminate the loop from within:

```
cout << "Guess a number between 1 and 10" << endl;
int number;
while (true) {
   cin >> number;
   if (number == 8)
       break;
}
cout << "You got it." << endl;</pre>
```

• Don't do this. It makes your code hard to read and debug.

Programming Assignment 4.5

- Rewrite PA3, Prepare a Lab Report, so that it uses a loop to enter the data for any number of rats (ask the user to specify the number of rats before the loop starts).
 - Then rewrite it to take the input from a file (do not input the number of rats, just loop until the end of the file).
- Rewrite PA4, Calculate a Cell Phone Bill, to ask the user if they want to repeat the program after the bill and savings are output. Also put the input validation in a loop.