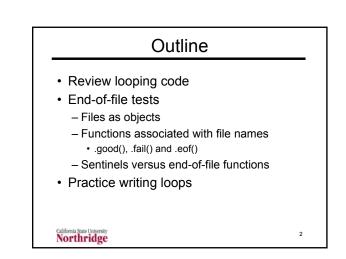
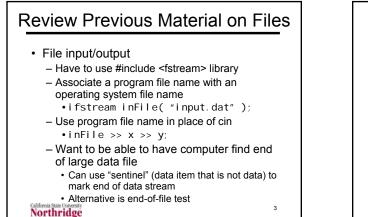
End-of-file Tests and Looping Summary

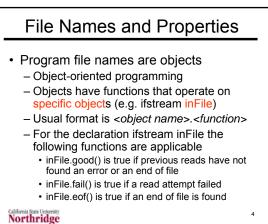
Larry Caretto Computer Science 106 Computing in Engineering and Science

March 16, 2006

California State University Northridge







Testing for End of File <EOF>Consider effects of three separate functions: .good(), .fail() and .eof() .good() is true if a future read statement may be possible (no error or end of file found yet) .fail() is true if a read statement could not be completed (some variables not read) .eof() is true if an end of file is found Where is the eof located? Important to understand EOF test

Possible <EOF> Locations

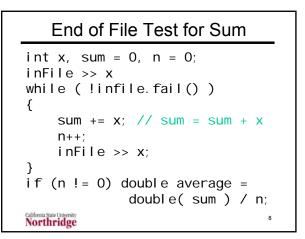
Example 1: 12 14 -23.2<EOF>

- Example 2: 12 14 -23.2 <EOF>
- Example 3: 12 14 -23.2<newline> <EOF>
- Example 1 file is saved immediately after the last digit is entered
- Example 2 file has spaces (but no newline after the last digit
- Example 3 file has <newline> (and possible spaces) after last digit

Northridge

6

Result	of in >	>> x >> y	′ >> Z;
Example 1:	12 14	-23.2 <eo< td=""><td>F></td></eo<>	F>
Example 2:	12 14	-23.2	<eof></eof>
Example 3:	12 14	-23.2 <nev< td=""><td>vline></td></nev<>	vline>
	<eof< td=""><td>></td><td></td></eof<>	>	
Function res	sults for	each sample	e file
Function:	in.good	() in.fail()	in.eof()
Example 1:	false	false	true
Example 2:	true	false	false
Example 3:	true	false	false
California State University			



End-of-file Exercise

- Read all the data from a file and determine the maximum, minimum, and number of data items on the file
- Hints
 - Use code similar to that on the last chart
 - Read the initial value and set the current minimum and maximum to that value
 - In the loop check each data item against the current minimum and maximum

9

11

California State University Northridge

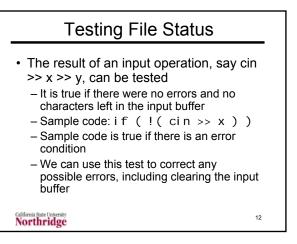
End of File Exercise Solution

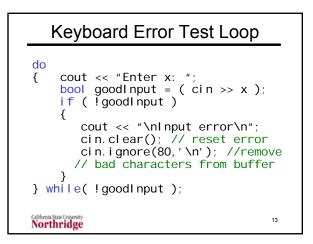
```
double x, xMin, xMax;
int count = 0;
ifstream inFile ( "input.dat" );
inFile >> x;
xMin = x;
xMax = x;
while (!infile.fail() )
{
    count++;
    if ( x > xMax )
        xMax = x;
    else if ( x < xMin )
        xMin = x;
    inFile >> x;
}
```

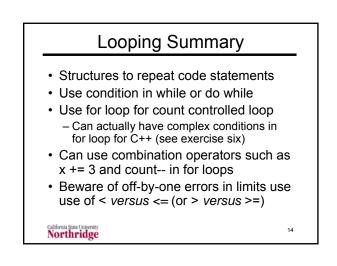
File Buffering

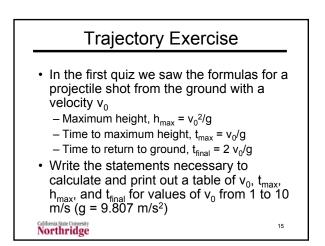
- Input and output information is placed in a buffer and transferred from input to code or code to output later
- Input transfer occurs when user presses the enter key
- If not all characters are read, the remaining characters are kept on the input buffer
 - Source of funny input results we saw in exercise two

California State University Northridge







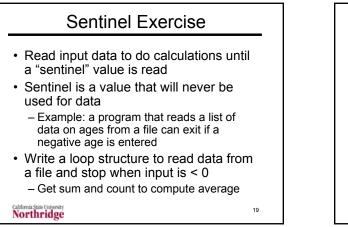


double v0, hMax, tMax, tfinal; const double g = 9.807; for (v0 = 1; v0 <= 10.5; v0++) { hMax = v0 * v0 / g; tMax = v0 / g; tFinal = 2 * v0 / g; cout << setw(4) << v0 << setw(9) << tMax << setw(9) << tMax << setw(9) << tFinal; }

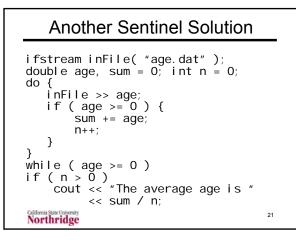
Another Trajectory Exercise

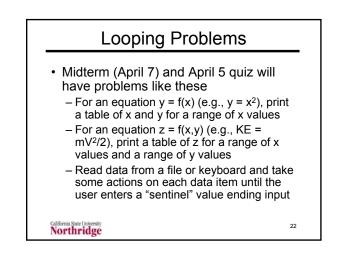
- The elevation above ground, z, for a particle shot from the ground at time = 0 with an initial velocity v_0 is given by the following equation $z = v_0 t gt^2/2$
- This equation is valid for $0 \le t \le 2v_0/g$
- Write the C++ code to calculate and print the elevation z as a function of time t so that there are 20 steps between t = 0 (when z = 0) and $t_{max} = 2v_0/g$ for input v_0

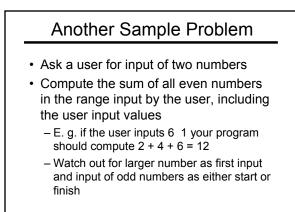
Another Exercise Solution



Sentinel Solution ifstream inFile("age.dat"); double age, sum = 0; int n = 0; inFile >> age; while (age >= 0) { sum += age; n++; inFile >> age; } if (n > 0) cout << "The average age is " << sum / n; </pre>

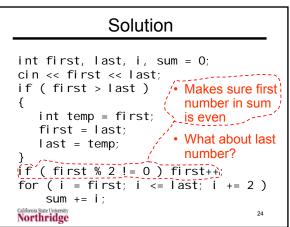






Northridge

23



Question about Solution

```
if ( first % 2 != 0 ) first++;
for ( i = first; i <= last; i += 2 )
    sum += i;</pre>
```

- The first statement assures that the initial number in the sum is even
- What about the last number?
 If last is even, the i <= last continuation condition will include it in the sum
- If last is odd the condition will include the last even number (i <= 7 includes 6, not 8)
 Colfernia State University