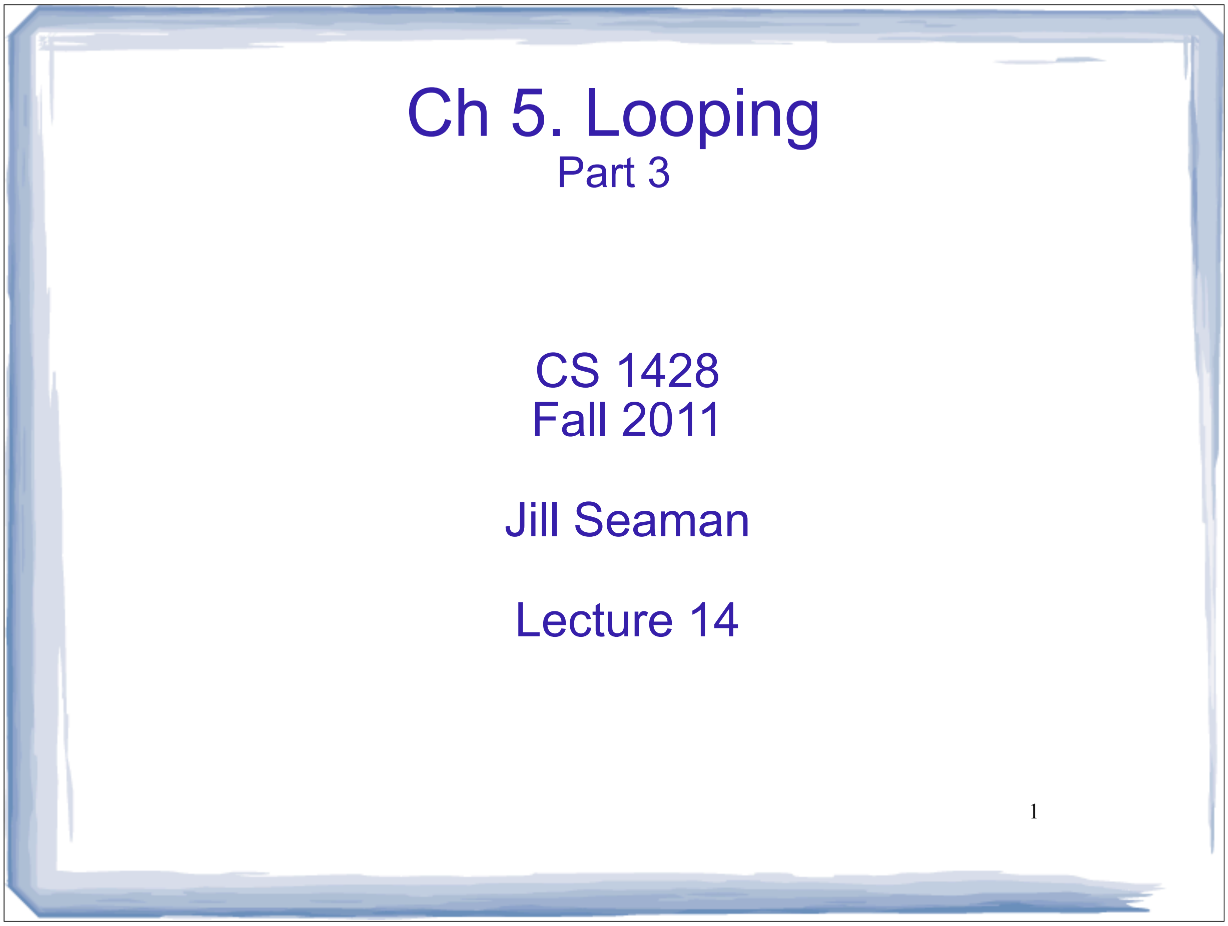


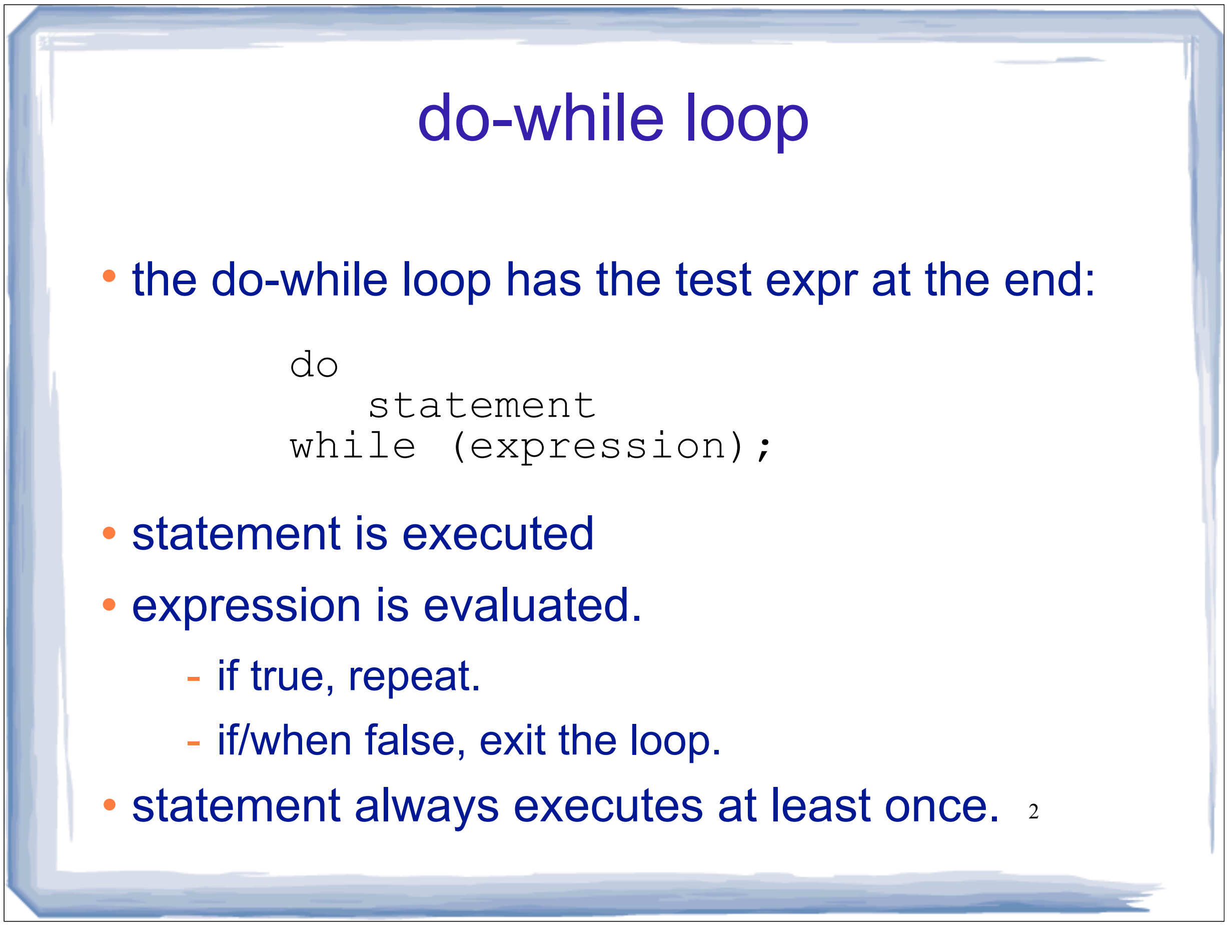
Ch 5. Looping

Part 3

CS 1428
Fall 2011

Jill Seaman

Lecture 14



do-while loop

- the do-while loop has the test expr at the end:

```
do
   statement
while (expression);
```

- statement is executed
- expression is evaluated.
  - if true, repeat.
  - if/when false, exit the loop.
- statement always executes at least once.

do while example

- Example:

```
int number = 1;
do
{
   cout << “Student” << number << endl;
   number++;
} while (number <= 3);

cout << “Done” << endl;
```

- Output:

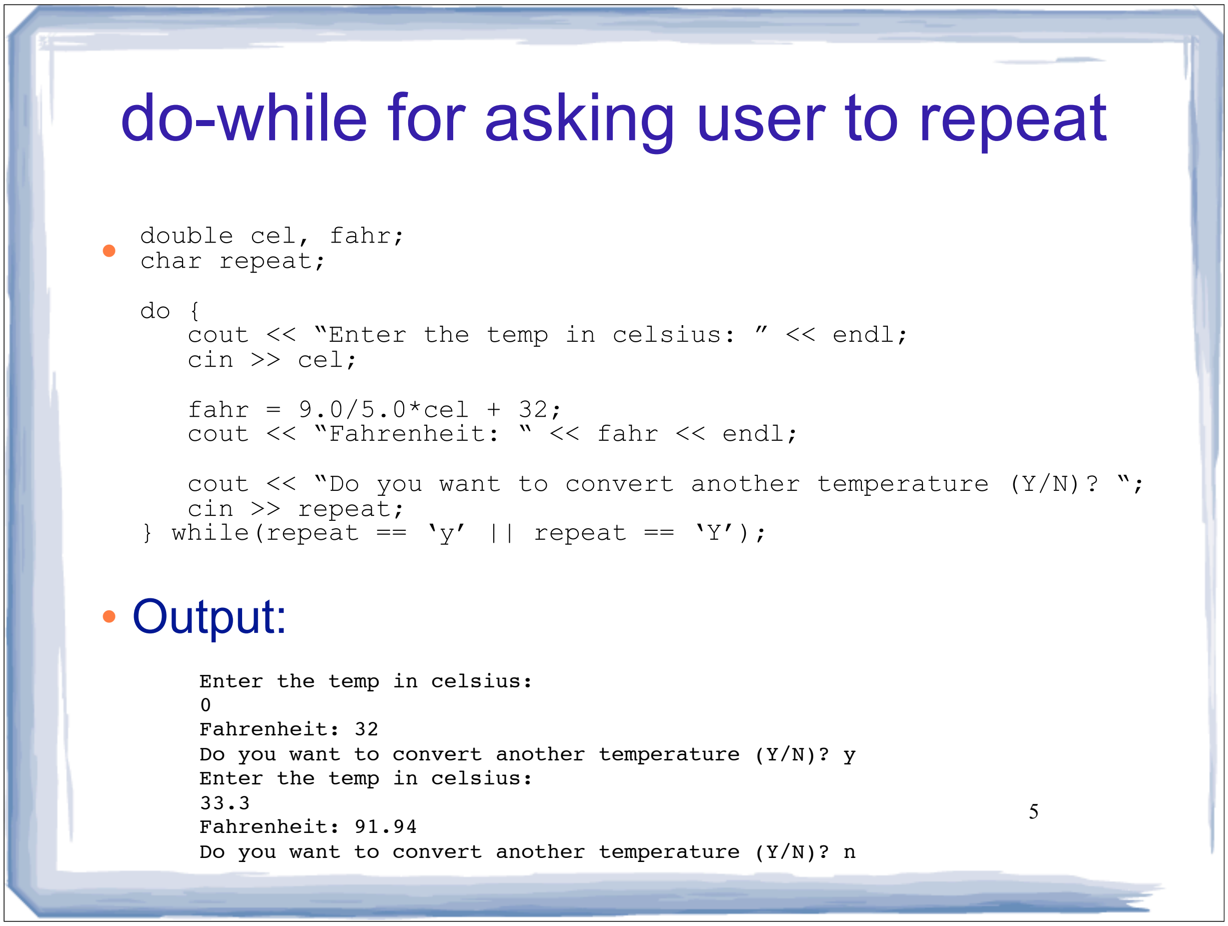
```
Student1
Student2
Student3
Done
```

Watch out

- What is output?

```
int x = 10;

do {
   cout << “Student” << x << endl;
   x++;
} while (x <= 3);

cout << “Done!” << endl;
```

- The body (statement) of the do-while is ALWAYS executed at least once, even if the test expression is false from the beginning.

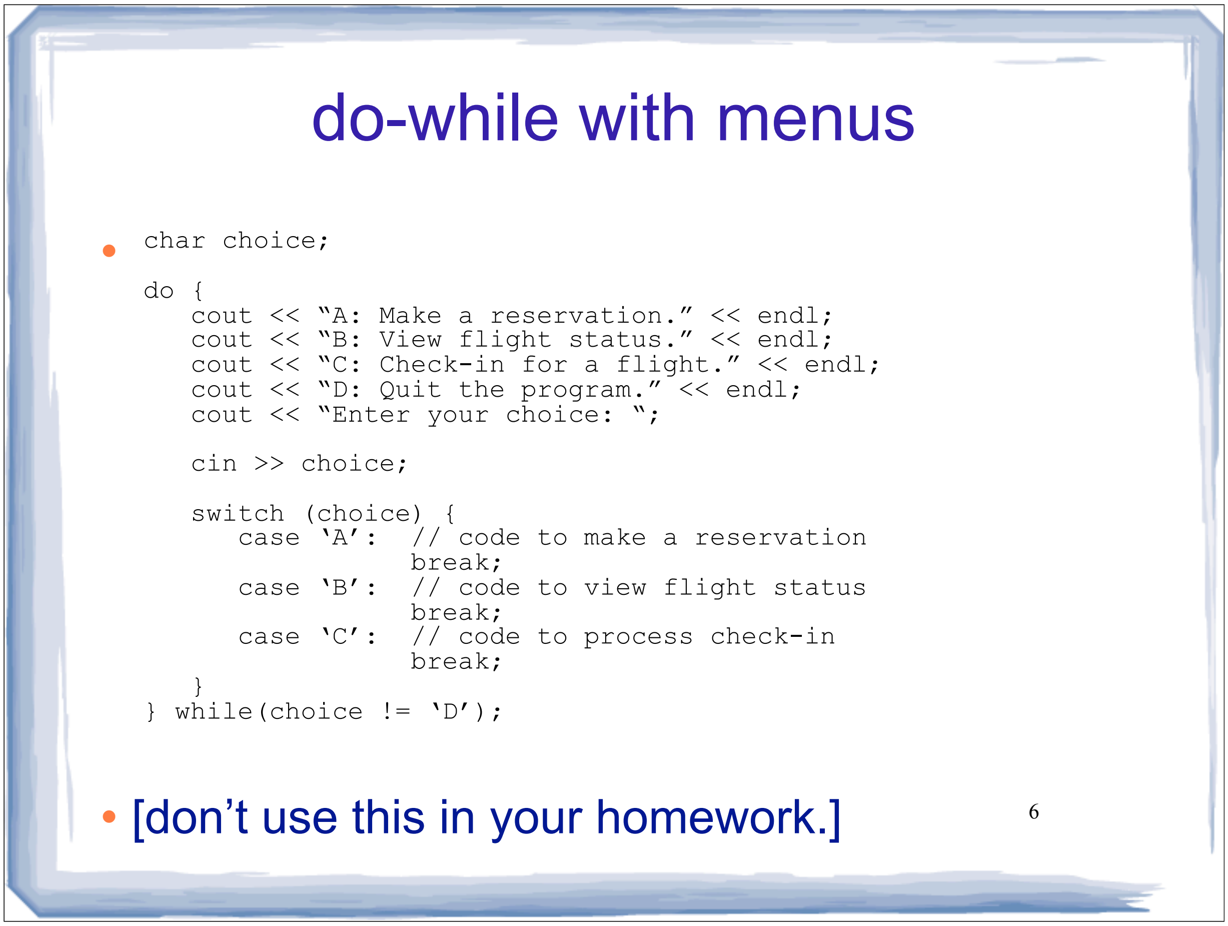


do-while for asking user to repeat

-
  ```
  double cel, fahr;
  char repeat;

  do {
     cout << “Enter the temp in celsius: ” << endl;
     cin >> cel;

     fahr = 9.0/5.0*cel + 32;
     cout << “Fahrenheit: “ << fahr << endl;

     cout << “Do you want to convert another temperature (Y/N)? “;
     cin >> repeat;
  } while(repeat == ‘y’ || repeat == ‘Y’);
  ```

- Output:

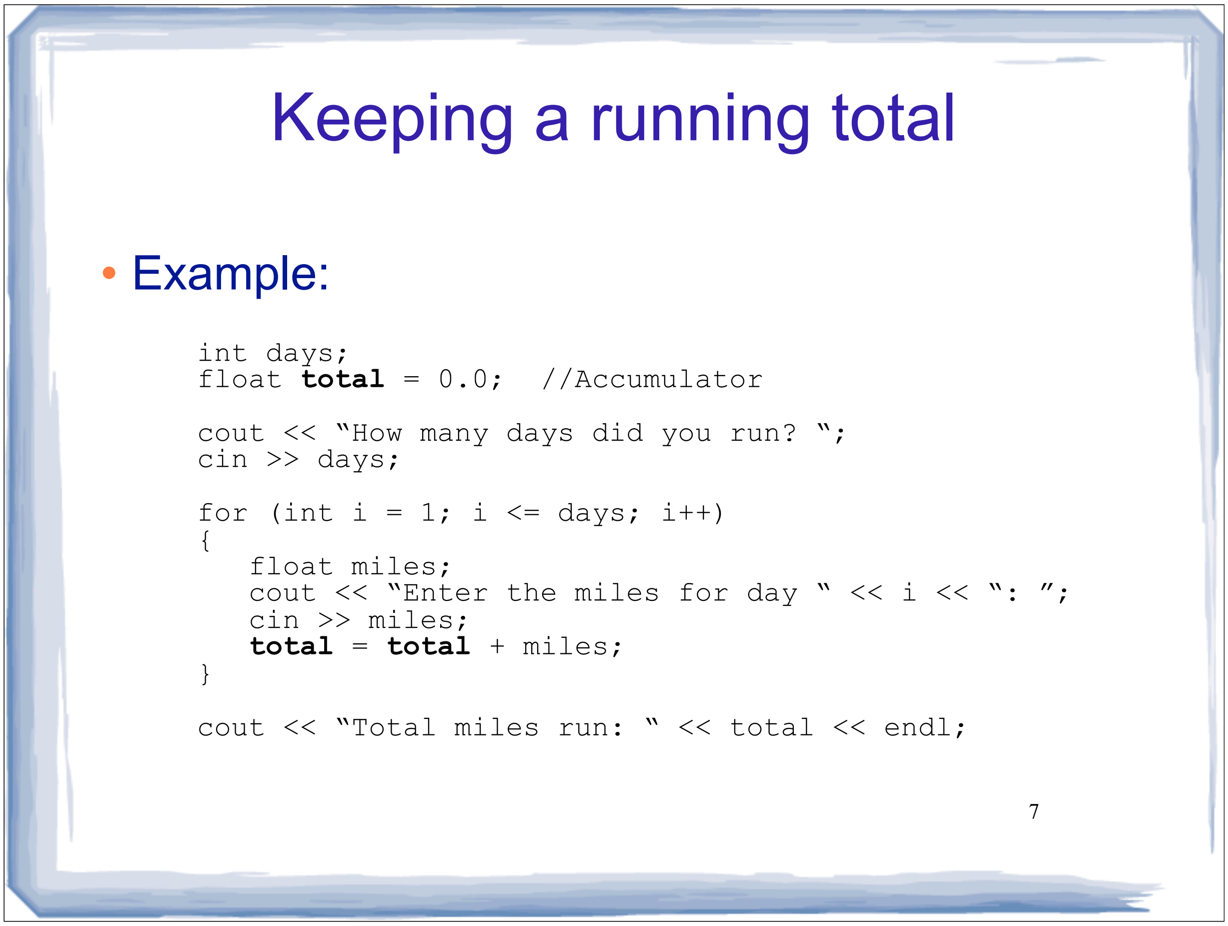
```
Enter the temp in celsius:
0
Fahrenheit: 32
Do you want to convert another temperature (Y/N)? y
Enter the temp in celsius:
33.3
Fahrenheit: 91.94
Do you want to convert another temperature (Y/N)? n
```



do-while with menus

-
  ```
  char choice;

  do {
     cout << “A: Make a reservation.” << endl;
     cout << “B: View flight status.” << endl;
     cout << “C: Check-in for a flight.” << endl;
     cout << “D: Quit the program.” << endl;
     cout << “Enter your choice: “;

     cin >> choice;

     switch (choice) {
        case ‘A’:  // code to make a reservation
                   break;
        case ‘B’:  // code to view flight status
                   break;
        case ‘C’:  // code to process check-in
                   break;
     }
  } while(choice != ‘D’);
  ```

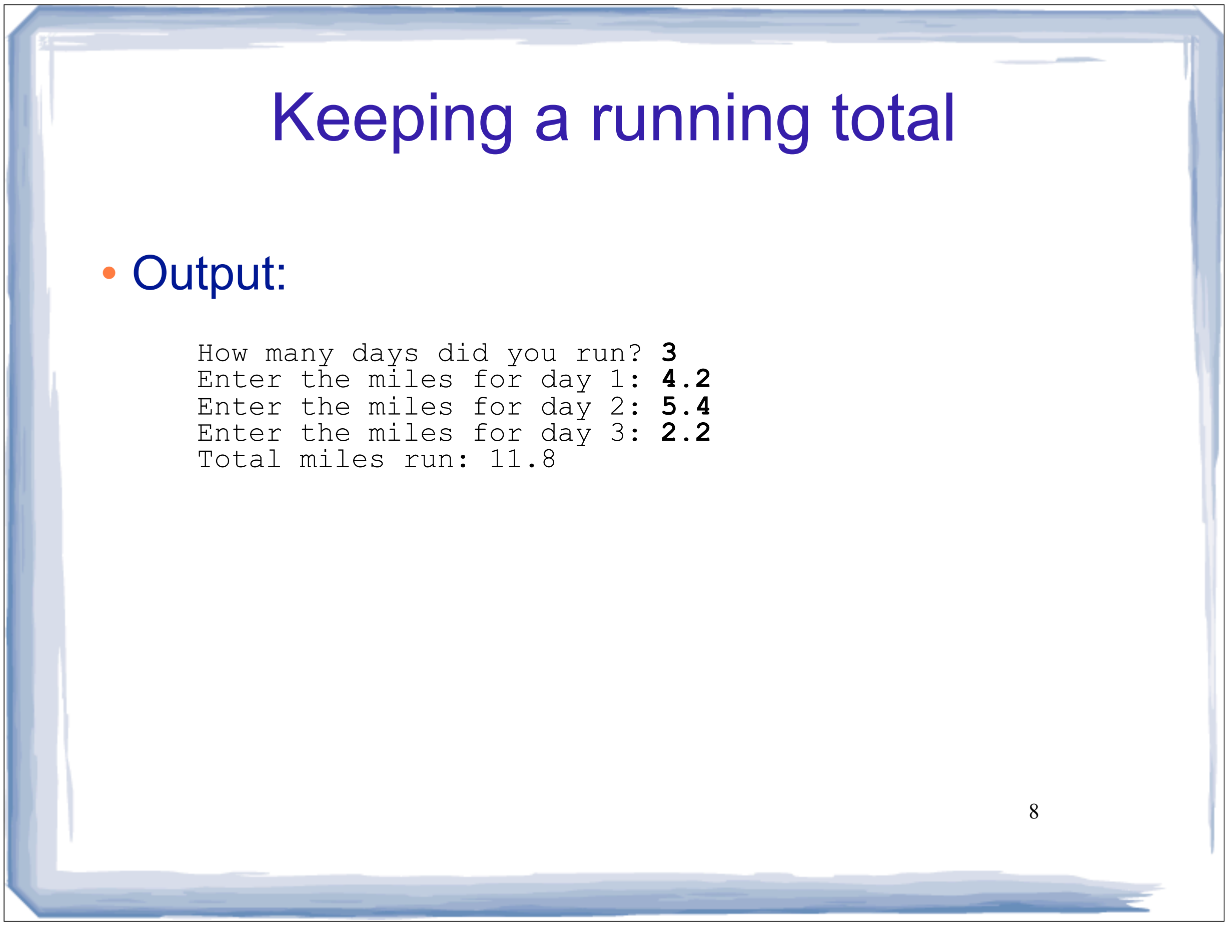
- [don’t use this in your homework.]



Keeping a running total

- Example:

```
int days;
float total = 0.0;  //Accumulator

cout << “How many days did you run? “;
cin >> days;

for (int i = 1; i <= days; i++)
{
   float miles;
   cout << “Enter the miles for day “ << i << “: ”;
   cin >> miles;
   total = total + miles;
}

cout << “Total miles run: “ << total << endl;
```



Keeping a running total

- Output:

```
How many days did you run? 3
Enter the miles for day 1: 4.2
Enter the miles for day 2: 5.4
Enter the miles for day 3: 2.2
Total miles run: 11.8
```

Sentinal controlled loop

- Use a special value to signify end of the data:

```
float total = 0.0;  //Accumulator
float miles;

cout << "Enter the miles you ran each day, ";
cout << "one number per line.\n";
cout << "Then enter -1 when finished.\n";

cin >> miles;

while (miles != -1)
{
   total = total + miles;
   cin >> miles;
}

cout << "Total miles run: " << total << endl;
```

Sentinal controlled loop

- Output:

```
Enter the miles you ran each day, one number per line.
Then enter -1 when finished.
4.2
5.4
2.2
-1
Total miles run: 11.8
```

- Sentinal value must NOT be a valid value (cannot run -1 miles).
- Requires a “priming read” before the loop starts,
    - so sentinal is NOT summed
    - loop can be skipped (if first value is -1)