CASE STUDY

5 Lightening Lanes

Problem Statement

On Tuesday afternoons, Lightening Lanes Bowling Alley runs a special class to teach children to bowl. Each lane has an instructor who works with a team of four student bowlers and instructs them as they bowl three lines (games). The management of Lightening Lanes has asked you to develop a program that will report each student's 3-game average score and compare it to the average score they bowled the previous week. In this way, the students can see how much they are improving. The program will utilize looping structures and data validation techniques learned in Chapter 5.

Input: Input should come from the keyboard, using suitable prompts. First the team name should be entered. Then, for each of the students on the team, the program should input their name, their previous week's average score, and their score from each of today's games. The score for each game must be between 0 and 300.

Output: Output should be written to a file. It should be in the form of a report that lists the team name, each student's name, their 3-game average from last week, and their 3-game average from today.

Program Design

General Pseudocode

The following general pseudocode lists the steps the program must carry out.

```
Open the output file
Input the team name
Print report heading which includes team name
For each of the students
   Input the student's name
   Input and validate their last week's bowling average
   Input, validate, and add up each of their game scores for today
   Compute their average for today
   Print student name, last week's average and today's average
End for
Close the file
```

Constant Variables

```
int NUM_STUDENTS = 4
    NUM_GAMES    = 3
    MAX_SCORE    = 300
```

Variables Whose Values Will be Input

```
string teamName            // the team name
       name                // a student's name
double oldAvg              // a student's previous week's average
       newAvg              // a student's average for this week
int    score               // a student's score for 1 game
```

Variable Whose Value Will be Accumulated

```
int total                  // a student's total score for today's games
```

Detailed Pseudocode (including actual variable names and needed calculations)

```
Open the output file
Input teamName
Print report heading which includes teamName
For each of the students
   total = 0
   Input name, oldAvg
   While oldAvg < 0 or oldAvg > MAX_SCORE
      Print error message
      Input oldAvg
   End While
   For each of today’s games
      Input score
      While score < 0 or score > MAX_SCORE
         Print error message
         Input score
      End While
      total = total + score
   End For
   newAvg = total / NUM_GAMES
   Print name, oldAvg, newAvg
End For
Close the file
```

The Program

The next step, after the pseudocode has been checked for logic errors, is to expand the pseudocode into the final program. The source code listing for this program follows.

LighteningLanes.cpp

```
// This program computes average scores for 3 games of bowling
// for each of 4 students. It displays this information in a file
// along with each student's name and previous average.
#include <iostream>
#include <iomanip>
#include <fstream>
#include <string>
using namespace std;

int main()
{
   const int NUM_STUDENTS = 4,
             NUM_GAMES = 3,
             MAX_SCORE = 300;

   ofstream report;       // output file stream object
   string teamName,       // the team name
          name;           // a student's name
   double oldAvg,         // a student's previous week's average
          newAvg;         // a student's average for today's games
   int    score,          // a student's score for 1 game
          total;          // a student's total 3-game score

   // Open file and print headings
   report.open ("bowling.rpt");
   cout << "Enter your team name: ";
   getline(cin, teamName);
   report << "\n   Weekly Bowling Report for " << teamName << "\n\n";
   report << "Name          Last week's avg   Today's avg\n\n";
   report << fixed << showpoint << setprecision(1);

   // Get and process information for each student
   for (int student = 1; student <= NUM_STUDENTS; student++)
   {   total =  0;       // Reset the accumulator to 0 for each student
       cout << "\nEnter your name: ";
       getline(cin, name);
       cout << "Enter your last week's average game score: ";
       cin  >> oldAvg;
       while ((oldAvg < 0) || (oldAvg > MAX_SCORE))
       {   cout << "Average must be between 0 and " << MAX_SCORE <<".\n"
                << "Please reenter your last week's average: ";
           cin  >> oldAvg;
       }
       for (int game = 1; game <= NUM_GAMES; game++)
       {   cout << "Enter today's score for game " << game << ": ";

```

```
           cin  >> score;
           while ((score < 0) || (score > MAX_SCORE))
           {   cout << "Average must be between 0 and " << MAX_SCORE <<".\n"
                    << "Reenter score for game " << game << ": ";
               cin  >> score;
           }
           total += score;
       }
       newAvg = static_cast<double>(total) / NUM_GAMES;
       report << left << setw(12) << name << right
              << setw(10) << oldAvg << setw(18) << newAvg << endl;

       cin.ignore();      // Clear \n from input buffer
   }
   report.close();
   return 0;
}
```

Program Screen Output with Example Input Shown in Bold

```
Enter your team name: the Mavericks[Enter]

Enter your name: Adam[Enter]
Enter your last week's average game score: 122[Enter]
Enter today's score for game 1: 121[Enter]
Enter today's score for game 2: 124[Enter]
Enter today's score for game 3: 131[Enter]

Enter your name: Ben[Enter]
Enter your last week's average game score: 119[Enter]
Enter today's score for game 1: 134[Enter]
Enter today's score for game 2: 131[Enter]
Enter today's score for game 3: 128[Enter]

Enter your name: Carol[Enter]
Enter your last week's average game score: 120[Enter]
Enter today's score for game 1: 118[Enter]
Enter today's score for game 2: 126[Enter]
Enter today's score for game 3: 130[Enter]

Enter your name: David[Enter]
Enter your last week's average game score: 131[Enter]
Enter today's score for game 1: 110[Enter]
Enter today's score for game 2: 138[Enter]
Enter today's score for game 3: 132[Enter]
```

Program Output Written to the Report File

```
    Weekly Bowling Report for the Mavericks

Name        Last week's avg      Today's avg

Adam           122.0                125.3
Ben            119.0                131.0
Carol          120.0                124.7
David          131.0                126.7
```