**C/C++ MOVIE RATINGS LAB REPORT**

**1) Enter your name, student ID, platform (Mac or PC) and date**

Name: Samuel Indurkar, 0888068

Class: CIS054 C/C++ Programming

Platform (Mac or PC): Mac and eclipse

Date: July 23 2017

**PROJECT DESCRIPTION:**

The project is described as PROGRAMMING PROJECTS #14 at the end of chapter 10 in the 8th Edition, pages 599-600.

The project is described as PROGRAMMING PROJECTS #11 at the end of chapter 10 in the 9th Edition, page 617.

A sample project is available on Canvas named AverageGrade that is very similar to the MovieRatings project, but it only is inputting one string (className) instead of both the movie name and the MPAA rating. Another difference is that the AverageGrade program scores grades A to F with point values from 4.0 to 0.0 (highest to lowest) while the MovieRatings scores the rating from 1 to 5 (lowest to highest). When using the AverageGrade program as a reference, make sure that the program for MovieRatings does not use variables such as Acount, Bcount, classSize, etc.

**LAB REPORT:  
2) Determine the Inputs, Processing and Outputs before creating the program**

|  |  |  |
| --- | --- | --- |
| **INPUTS** | **PROCESSING** | **OUTPUTS** |
| none from user, but I provided three pre-coded input objects with Movie-name, MPAA-rating, and ratings | Create the object. default constructor assigns the movie-name and MPAA-rating. Each voter increments the total vote-count, and also increments the appropriate rating-count. | Display the movie name, MPAA rating and the average rating |

**LAB REPORT:**

**3) Fill in the TEST VALUES & RESULTS table**Fill in the **Test Data Values** and **Expected Results** as shown on the lab assignment from Canvas. Your test data must show at least three movies with different MPAA ratings. The number of people rating the movies for the different categories (1-5) must be different for each category and different for each movie.  
Fill in the **Actual Results** after you have run and tested your program

|  |  |  |
| --- | --- | --- |
| **TEST DATA VALUES**  Number of people who rate the movie as 1,2,3,4 and 5 | **EXPECTED RESULT**  Computed values before the program is run | **ACTUAL RESULT**  Fill in the output displayed  by the program |
| Moana, PG-13,  5,4,3,2,1 | Average should be 3 | The average rating for MOANA is 3 and the MPAA rating is PG-13 |
| SUPERMAN-2, G,  5,1,2,4,5,4 | Average displayed should be 3 | The average rating for SUPERMAN-2 is 3 and the MPAA rating is G |
| CARS3, PG,  5,5,5,4,5,4,5,4 | Average displayed should be 4 | The average rating for CARS3 is 4 and the MPAA rating is PG |

**DISCUSSION:**

**4) Complete the DISCUSSION section. It does not need to be long, but it needs to be complete.**4a) What did you do to develop the program? ("Followed the Directions" is not a complete description)

I provided three pre-coded input objects with Movie-name, MPAA-rating, and ratings. Created the object using the default constructor. The default constructor assigns the movie-name and MPAA-rating. Each voter increments the total vote-count, and also increments the appropriate rating-count. At the end, Display the movie name, MPAA rating and the average rating

4b) What problems did you have and how did you overcome the problems?

I had the usual problems of naming issues, for example in a few spots i called “totalvotes” and in others i called “totalVotes”. The other problem I had was: MOVIE1 was created with the default 0 constructor and it used setName however the MOVIE2 used the constructor with 2 arguments but when i created the object i only put one argument. So, the compiler was giving me compile error which went away after i provided 2 args to the constructor.

**PROGRAM OUTPUT:**

**5) Show screen shots that include at least three different movies with different MPAA values and different rating scores.**

Refer to previous lab assignments for instructions on how to capture a screen or portions of a screen for either the PC or a Mac

****

**PROGRAM LISTING:**

**6) Copy and paste the code that YOU typed to make the program work. Your program should include a comment block at the top that shows the name of the program, date, version and your name.**

/\*

\* movie\_rating.h

\*

\* Created on: Jul 23, 2017

\* Author: Samuel

\*/

#ifndef MOVIERATING\_h // only compile this header file one time

#define MOVIERATING\_h

class movie\_rating

{

private:

char movieName[100];

char MPAARating[20];

int rated\_1;

int rated\_2;

int rated\_3;

int rated\_4;

int rated\_5;

int totalVotes;

public:

// constructors

movie\_rating(); // default constructor

movie\_rating(char \*name, char \*MPRat); // constructor with one argument

void addRating(int ratingNumber);

void setName(char \*name); // mutator - sets the name into the object

char \*getName(); // accessor - gets the name from the object

void setMPAARating(char \*rating); // mutator - sets the name into the object

char \*getMPAARating(); // accessor - gets the name from the object

double getAverage();

};

#endif

/\*

\* movie\_rating.cpp

\*

\* Created on: Jul 23, 2017

\* Author: Samuel

\*/

#include "movie\_rating.h"

#include <cstring>

#include <cctype>

#include <iostream>

using namespace std;

movie\_rating::movie\_rating() // default constructor

{

movieName[0] = '\0'; // set to an empty string

MPAARating[0] = 0; // set to an empty string

rated\_1 = rated\_2 = rated\_3 = rated\_4 = rated\_5 = 0;

totalVotes = 0;

}

movie\_rating::movie\_rating(char \*name, char \*MPRat)

{

strcpy (movieName, name); // use if strcpy is required by the compiler

strcpy (MPAARating, MPRat);

rated\_1 = rated\_2 = rated\_3 = rated\_4 = rated\_5 = 0;

totalVotes = 0;

}

void movie\_rating::addRating(int ratingNumber)

{

switch ( ratingNumber )

{

case 1:

rated\_1++; // count the number of 1's

totalVotes++; // count the number of students

break;

case 2:

rated\_2++; // count the number of 2's

totalVotes++; // count the number of students

break;

case 3:

rated\_3++; // count the number of 3's

totalVotes++; // count the number of students

break;

case 4:

rated\_4++; // count the number of 4's

totalVotes++; // count the number of students

break;

case 5:

rated\_5++; // count the number of 5's

totalVotes++; // count the number of students

break;

default:

cout << ratingNumber << " was an illegal input value" << endl;

}

}

void movie\_rating::setName(char\* name) // mutator - sets the movieName

{

strcpy (movieName, name); // use if strcpy is required by the compiler

}

char\* movie\_rating::getName() // accessor - gets the movieName

{

return movieName;

}

void movie\_rating::setMPAARating(char\* rating) // mutator - sets the movieName

{

strcpy (MPAARating, rating); // use if strcpy is required by the compiler

//strcpy\_s (movieName, 100, name); // use if strcpy\_s is required by the compiler

}

char\* movie\_rating::getMPAARating() // accessor - gets the movieName

{

return MPAARating;

}

double movie\_rating::getAverage()

{

int average;

average = (rated\_1 \* 1.0

+ rated\_2 \* 2.0

+ rated\_3 \* 3.0

+ rated\_4 \* 4.0

+ rated\_5 \* 5.0)

/ totalVotes;

return (average);

}

/\*

\* movie\_rating\_main.cpp

\*

\* Created on: Jul 23, 2017

\* Author: Samuel

\*/

#include "movie\_rating.h"

#include <iostream>

using namespace std;

int main(int argc, char\* argv[])

{

movie\_rating MOVIE1; // create an object of class movie\_rating

MOVIE1.setName("MOANA"); // use mutator to set the class name

MOVIE1.setMPAARating("PG-13"); // use mutator to set the class name

// 5 people rated this movie

MOVIE1.addRating(5);

MOVIE1.addRating(4);

MOVIE1.addRating(3);

MOVIE1.addRating(2);

MOVIE1.addRating(1);

cout << "The average rating for " << MOVIE1.getName() << " is " << MOVIE1.getAverage() << " and the MPAA rating is " << MOVIE1.getMPAARating()<< endl;

movie\_rating MOVIE2("SUPERMAN-2", "G"); // create an object of class movie\_rating

// 6 people rated this movie

MOVIE2.addRating(5);

MOVIE2.addRating(1);

MOVIE2.addRating(2);

MOVIE2.addRating(4);

MOVIE2.addRating(5);

MOVIE2.addRating(4);

cout << "The average rating for " << MOVIE2.getName() << " is " << MOVIE2.getAverage() << " and the MPAA rating is " << MOVIE2.getMPAARating()<< endl;

movie\_rating MOVIE3("CARS3", "PG"); // create an object of class movie\_rating

// 8 people rated this movie

MOVIE3.addRating(5);

MOVIE3.addRating(5);

MOVIE3.addRating(5);

MOVIE3.addRating(4);

MOVIE3.addRating(5);

MOVIE3.addRating(4);

MOVIE3.addRating(5);

MOVIE3.addRating(4);

cout << "The average rating for " << MOVIE3.getName() << " is " << MOVIE3.getAverage() << " and the MPAA rating is " << MOVIE3.getMPAARating()<< endl;

return 0;

}