



MOVIE WATCHLIST

DONE BY

23X031 - PONSIVANESH

23X042 - SANJIVKUMAR

SOURCE CODE

```
#include <iostream>

#include <fstream>

#include <string>

#include <algorithm>

#include <iomanip> // For setw

using namespace std;

// ANSI escape code for green color

const string GREEN_COLOR_CODE = "\033[1;32m";

// ANSI escape code for resetting color

const string RESET_COLOR_CODE = "\033[0m";


const int MAX_MOVIES = 100;
```

```
struct Movie {  
  
    string title;  
  
    string genre;  
  
    double rating;  
  
};
```

```
struct Watchlist {  
  
    Movie movies[MAX_MOVIES];  
  
    int count = 0;  
  
    string recentlyPlayed;  
  
    string lastPlayed;  
  
};
```

```
void addMovie(Watchlist& watchlist, const Movie& movie) {  
  
    if (watchlist.count < MAX_MOVIES) {  
  
        watchlist.movies[watchlist.count++] = movie;  
  
        ofstream file("watchlist.txt", ios::app);  
  
        file << movie.title << "|" << movie.genre << "|" << fixed << setprecision(1) <<  
movie.rating << endl;  
  
        cout << GREEN_COLOR_CODE << "---- Movie '" << movie.title << "' added to the  
watchlist. ----" << RESET_COLOR_CODE << endl;  
  
    } else {  
  
        cout << GREEN_COLOR_CODE << "---- Watchlist is full. Cannot add more movies. ----"
```

```
<< RESET_COLOR_CODE << endl;

}

}
```

```
void deleteMovie(Watchlist& watchlist, const string& movieTitle) {

    for (int i = 0; i < watchlist.count; ++i) {

        if (watchlist.movies[i].title == movieTitle) {

            for (int j = i; j < watchlist.count - 1; ++j) {

                watchlist.movies[j] = watchlist.movies[j + 1];

            }

            --watchlist.count;

            cout << GREEN_COLOR_CODE << "---- Movie '" << movieTitle << "' removed from the  
watchlist. ----" << RESET_COLOR_CODE << endl;

            return;

        }

    }

    cout << GREEN_COLOR_CODE << "---- Movie '" << movieTitle << "' not found in the  
watchlist. ----" << RESET_COLOR_CODE << endl;

}
```

```
void displayWatchlist(const Watchlist& watchlist) {

    if (watchlist.count == 0) {

        cout << GREEN_COLOR_CODE << "---- Watchlist is empty. ----" <<  
RESET_COLOR_CODE << endl;

        return;

    }

}
```

```

}

cout << GREEN_COLOR_CODE << "---- Watchlist: ----" << RESET_COLOR_CODE << endl;

cout << setw(30) << "Title" << setw(20) << "Genre" << setw(10) << "Rating" << endl;

for (int i = 0; i < watchlist.count; ++i) {

    cout << setw(30) << watchlist.movies[i].title << setw(20) << watchlist.movies[i].genre <<
    setw(10) << fixed << setprecision(1) << watchlist.movies[i].rating << endl;

}

}

```

```

void searchMovie(const Watchlist& watchlist, const string& movieTitle) {

    for (int i = 0; i < watchlist.count; ++i) {

        if (watchlist.movies[i].title == movieTitle) {

            cout << GREEN_COLOR_CODE << "---- Movie '" << movieTitle << "' found in the
watchlist. ----" << RESET_COLOR_CODE << endl;

            return;

        }

    }

    cout << GREEN_COLOR_CODE << "---- Movie '" << movieTitle << "' not found in the
watchlist. ----" << RESET_COLOR_CODE << endl;

}

```

```

void playMovie(Watchlist& watchlist, const string& movieTitle) {

    for (int i = 0; i < watchlist.count; ++i) {

        if (watchlist.movies[i].title == movieTitle) {

```

```
    cout << GREEN_COLOR_CODE << "---- Now playing: " << movieTitle << " ----" <<
RESET_COLOR_CODE << endl;
```

```
    watchlist.recentlyPlayed = movieTitle;
```

```
    watchlist.lastPlayed = movieTitle;
```

```
    return;
```

```
}
```

```
}
```

```
    cout << GREEN_COLOR_CODE << "---- Movie '" << movieTitle << "' not found in the
watchlist. ----" << RESET_COLOR_CODE << endl;
```

```
}
```

```
void sortWatchlist(Watchlist& watchlist) {
```

```
    sort(watchlist.movies, watchlist.movies + watchlist.count, [](const Movie& a, const
Movie& b) {
```

```
        return a.title < b.title;
```

```
    });
```

```
    cout << GREEN_COLOR_CODE << "---- Watchlist sorted by title. ----" <<
RESET_COLOR_CODE << endl;
```

```
}
```

```
void sortRecentlyPlayed(Watchlist& watchlist) {
```

```
    cout << GREEN_COLOR_CODE << "---- Recently Played: " << watchlist.recentlyPlayed <<
" ----" << RESET_COLOR_CODE << endl;
```

```
}
```

```
void displayLoadedMovies(Watchlist& watchlist) {
```

```
    cout << GREEN_COLOR_CODE << "---- Loaded Movies: ----" << RESET_COLOR_CODE <<  
endl;
```

```
    for (int i = 0; i < watchlist.count; ++i) {
```

```
        cout << GREEN_COLOR_CODE << watchlist.movies[i].title << RESET_COLOR_CODE  
<< endl;
```

```
    }
```

```
}
```

```
void clearWatchlist(Watchlist& watchlist) {
```

```
    watchlist.count = 0;
```

```
    cout << GREEN_COLOR_CODE << "---- Watchlist cleared. ----" << RESET_COLOR_CODE  
<< endl;
```

```
}
```

```
void menu(Watchlist& watchlist) {
```

```
    int choice;
```

```
    string movieTitle;
```

```
    do {
```

```
        cout << GREEN_COLOR_CODE << "\n---- Menu: ----\n"
```

```
            "1. Add Movie\n"
```

```
            "2. Delete Movie\n"
```

```
            "3. Display Watchlist\n"
```

"4. Search Movie\n"

"5. Play Movie\n"

"6. Sort Watchlist\n"

"7. Sort Recently Played\n"

"8. Display Loaded Movies\n"

"9. Clear Watchlist\n"

"10. Total Movies\n"

"11. Recently Played Movie\n"

"12. Last Played Movie\n"

"13. Exit\n"

"Enter your choice: " << RESET_COLOR_CODE;

cin >> choice;

switch (choice) {

case 1:

{

Movie movie;

cout << GREEN_COLOR_CODE << "---- Enter Movie Title: " <<
RESET_COLOR_CODE;

cin >> movie.title;

cout << GREEN_COLOR_CODE << "---- Enter Genre: " <<
RESET_COLOR_CODE;

cin >> movie.genre;

cout << GREEN_COLOR_CODE << "---- Enter Rating (0.0 - 10.0): " <<

RESET_COLOR_CODE;

cin >> movie.rating;

addMovie(watchlist, movie);

}

break;

case 2:

cout << GREEN_COLOR_CODE << "---- Enter the title of the movie you want to delete: " << RESET_COLOR_CODE;

cin >> movieTitle;

deleteMovie(watchlist, movieTitle);

break;

case 3:

displayWatchlist(watchlist);

break;

case 4:

cout << GREEN_COLOR_CODE << "---- Enter the title of the movie you want to search: " << RESET_COLOR_CODE;

cin >> movieTitle;

searchMovie(watchlist, movieTitle);

break;

case 5:

cout << GREEN_COLOR_CODE << "---- Enter the title of the movie you want to play: " << RESET_COLOR_CODE;

cin >> movieTitle;

playMovie(watchlist, movieTitle);

break;

case 6:

sortWatchlist(watchlist);

break;

case 7:

sortRecentlyPlayed(watchlist);

break;

case 8:

displayLoadedMovies(watchlist);

break;

case 9:

clearWatchlist(watchlist);

break;

case 10:

**cout << GREEN_COLOR_CODE << "---- Total Movies: " << watchlist.count << " ----"
<< RESET_COLOR_CODE << endl;**

break;

case 11:

**cout << GREEN_COLOR_CODE << "---- Recently Played Movie: " <<
watchlist.recentlyPlayed << " ----" << RESET_COLOR_CODE << endl;**

break;

case 12:

**cout << GREEN_COLOR_CODE << "---- Last Played Movie: " <<
watchlist.lastPlayed << " ----" << RESET_COLOR_CODE << endl;**

```

        break;

    case 13:

        cout << GREEN_COLOR_CODE << "---- Exiting... ----" << RESET_COLOR_CODE <<
endl;

        break;

    default:

        cout << GREEN_COLOR_CODE << "---- Invalid choice. Please try again. ----" <<
RESET_COLOR_CODE << endl;

    }

} while (choice != 13);

}

```

```

int main() {

    cout << GREEN_COLOR_CODE << "*" WELCOME TO MOVIE WATCHLIST *" <<
RESET_COLOR_CODE << endl;

    Watchlist watchlist;

    menu(watchlist);

    return 0;

}

```

OUTPUT

```
1. Add Movie
2. Delete Movie
3. Display Watchlist
4. Search Movie
5. Play Movie
6. Sort Watchlist
7. Sort Recently Played
8. Display Loaded Movies
9. Clear Watchlist
10. Total Movies
11. Recently Played Movie
12. Last Played Movie
13. Exit
Enter your choice: 1
---- Enter Movie Title: LEGEND
---- Enter Genre: COMEDY
---- Enter Rating (0.0 - 10.0): 10
---- Movie 'LEGEND' added to the watchlist. ----
```

```
---- Menu: ----
1. Add Movie
2. Delete Movie
3. Display Watchlist
4. Search Movie
5. Play Movie
6. Sort Watchlist
7. Sort Recently Played
8. Display Loaded Movies
9. Clear Watchlist
10. Total Movies
11. Recently Played Movie
12. Last Played Movie
13. Exit
```

Enter your choice: 3

---- Watchlist: ----

Title	Genre	Rating
LEO	ACTION	8.6
PREMALU	LOVE	9.0
PIZZA	HORROR	7.0
LEGEND	COMEDY	10.0