

## Lab 6: Movie Rating

### OBJECTIVE

This lab aims to make you familiar with concepts of structs, templates and function overloading. You will also use multiple files for this lab

### ASSIGNMENT: RATE THE MOVIES

Write a program that rates two movies (**PG-13 or lower**) on a scale of 10 and prints it using a template function.

### SAMPLE OUTPUT

Welcome to the movie rating system.

Enter Movie 1 details:

Movie Title : Spectre

Lead : Daniel Craig

On a scale of 1 to 10, rate the actor

Based on “acting ability”: some integer > 0

Based on “appearance” : some integer > 0

Based on “popularity” : some integer > 0

On a scale of 1 to 10, rate the plot

Based on “freshness”: some integer > 0

Based on “execution”: some integer > 0

\*\*\*\*\*

Enter Movie 2 details

Movie Title : The Dark Knight

Lead : Christian Bale

On a scale of 1 to 10, rate the actor

Based on “acting ability”: some integer > 9

Based on “appearance” : some integer > 9

Based on “popularity” : some integer > 0

On a scale of 1 to 10, rate the plot

Based on “freshness”: some integer > 9

Based on “execution”: some integer > 9

\*\*\*\*\*

Actor rating for Spectre is: 6/10

Actor rating for The Dark Knight is: 9/10

Plot rating for Spectre is: 8/10

Plot rating for The Dark Knight is: 9/10

Overall rating for Spectre is: 6.0/10  
Overall rating for The Dark Knight is: 9.0/10

## FUNCTION PROTOTYPES

The following are sample function prototypes:

```
void getInput(Movie &movie);
```

```
int calcRating(int actorParam1, int actorParam2, int actorParam3);
```

```
int calcRating(int plotParam1, int plotParam2);
```

```
void setOverallRating(Movie &movie);
```

## FUNCTION SAMPLE

```
struct Movie
{
    string title;
    float overallRating;
    int actorRating;

    . . .

    int actorParam1, actorParam2, actorParam3;
    int plotParam1, plotParam2;
};
```

## TEMPLATE SAMPLE

```
template <typename T1, typename T2, typename T3>
void printRatings (T1 param1, T2 param2, T3 param3)
{
    // Use of param1 , param2 , param3;
}
```

## HINTS

- Use `getline( cin, varName)→` to get more than one string of input.
- Try to use the same/similar function names as suggested
- **Templates will not be prototyped.** However, definition goes in the header.cpp
- Actor and plot ratings remain integers at all time. Overall rating is a float.

#### GRADES WILL BE BASED ON

- Multiple files, templates, structs and function overloading.
- Appropriate use of pass by value and pass by reference.
- Use of appropriate loops for input checking. Minimized code redundancy.
- Function modularity. eg. only printRatings() will print the ratings
- Brilliance of coding logic. Case in point: Rounding of integer ratings to integer values.
- No pointers or global variables.
- Use sample output **structure** only.
- Completing on time.
- Readability of the program, using comment header.
- Correctness of the program. b
- Following programming conventions.

#### SUBMIT YOUR WORK

- To submit your work: **cssubmit 1580 f 8** (4:00 p.m. to 5:50 p.m.)  
**cssubmit 1580 g 8** (6:00 p.m. to 7:50 p.m.)