CS 1101 - A-term 16

Homework 1 - Structs

Due: Wednesday, Sept 7 at 5pm

Read the expectations on homework. You may do this homework and all subsequent homeworks with your homework partner.

Assignment Goals

- To make sure you can define structs
- To make sure you can write functions using conditions, structs, and nested structs
- To make sure you can develop a set of test cases for a function

---

Reminders

- Your solutions to the following problems should use helper functions in place of repeated code or to improve the readability of your code.
- Remember to include a signature, purpose, and test cases for every function you write, including helpers.
- Make sure you read each problem carefully and create a signature that conforms to the problem description. *You must name each function with the exact name specified in the problem. Your signature must conform to the problem description. Otherwise, we won't be able to run our automated tester on your program, and you'll lose points*. Programs that don't work with our auto-tester (and thus must be tested manually) will be penalized with a deduction of 10% of the total number of points for the assignment.
- Each test case you develop should be annotated with a brief comment that describes the situation being tested.

The Assignment

The website for a movie theater contains information about each of the films currently being shown in the theater. The following information is stored for a film:

- the title of the film
- the film's genre (drama, comedy, family, etc.)
- the film's rating. A rating can be one of G, PG, PG-13, R, NC-17, NR.
- the running time of the film, in minutes
- the date the film opened at the theater (it should include the year, month, and day)
- the total box office receipts collected so far for the film (in millions of dollars)

Problems

1. Provide data definitions for both a Film and a Date. Include both the define-structs and at least three examples for each data definition. When creating a struct, the order of the fields in the constructor should match the order given in the descriptions above. The names of the fields in the struct don't matter, but the order does. For example, the fields of the struct for a film should be the title, genre, rating, running time,

opening date, and receipts collected, in that order. *Failure to define the fields in the given order will cause our auto-tester to fail, and you will lose points*. Your struct for a date should have 3 fields, one for the year, one for the month, and one for the day, in that order. Each field in a date is of type Natural.

2. In a comment in the Defintions Window, state the *signatures* of all the operators that are created by Racket for your Film struct.

3. Write a function `suitable-for-children?` which consumes a Film and returns true if the rating of the film is G, PG, or PG-13, and returns false otherwise.

4. Write a function `difference-in-receipts`. The function consumes two films and produces a Number. The number produced is the difference between the box office receipts for the two films (the result should be a non-negative number).

5. Write a function `modify-rating` which consumes a Film and a String, and produces a Film. The film that is produced is the same as the original except that the film's rating has been replaced by the given rating.

6. Write a function `opens-before?`, which consumes a Film and a Date, and produces a Boolean. The function produces true if the given film opens before the given date, and returns false otherwise.

Grading

Here is the grading rubric that the graders will use for Homework 1.

What to Turn In

Submit your `.rkt` file to InstructAssist. Follow the naming conventions for homework files. BOTH partners' names and wpi ccc usernames MUST be listed in a comment at the beginning of your file (you will lose points if these rules are not followed). Programs are due at 5pm on Wednesday, September 7. Late programs will be accepted until 5pm on Thursday, September 8. Programs will not be accepted for submission after 5pm on Thursday, September 8. **No exceptions.**