

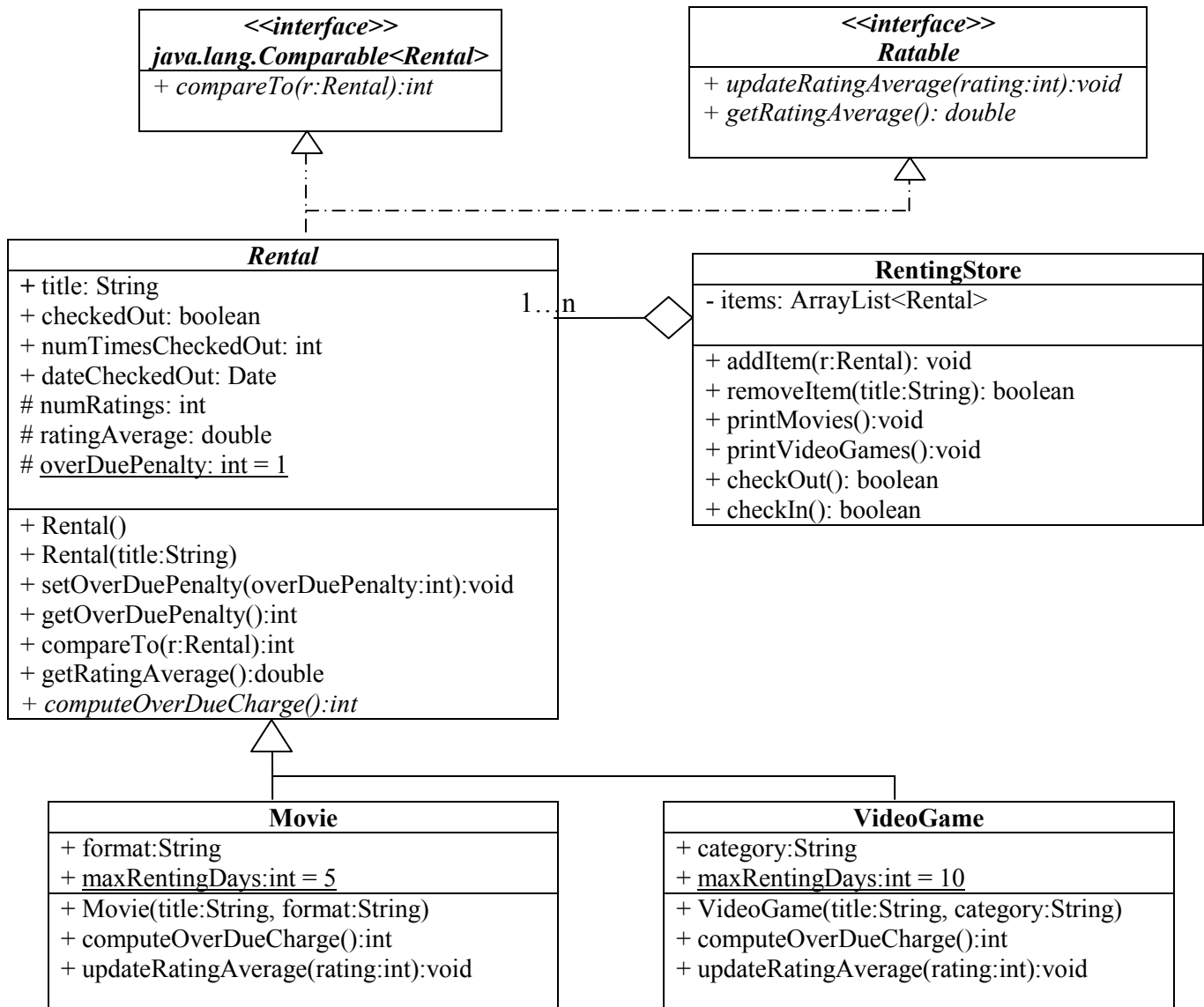


**University Of Jordan
Computer Engineering Department**

**Object-Oriented Problem Solving Lab
Final Exam – Spring 2014
Eng. Asma Abdel Karim**

Name: University ID: Computer Number:

- ❖ Write a JAVA program to build an appropriate implementation for the UML diagram below, which models a movie and video games renting store, using the concepts of Object Oriented Programming that you learned.



❖ Note the following:

- **In addition to the methods shown in the previous UML diagram, you are required to:**

➤ Override the *toString* method of the *Object* class in classes *Rental*, *Movie*, and *VideoGame*, as follows:

- **In class *Rental***, the *toString* method should return a String that consists of:

1. The rental *title*, then a *comma*, then
2. If the rental is checked out the statement “*checked out on (date in which it was checked out)*”, if the rental is not checked out the statement “*not checked out*”, then a *comma*, then
3. The statement: *number of times checked out: _____*, then a *comma*, then
4. The statement: *average rating: _____*.

- **In class *Movie***, the *toString* method should return the string returned by the *toString* method of the *Rental* class in addition to a *comma* and the statement: *available on _(format)_*.

- **In class *VideoGame***, the *toString* method should return the string returned by the *toString* method of the *Rental* class in addition to a *comma* and the statement: *category : __(category)__*.

➤ Override the *equals* method of the *Object* class in class *Rental*, such that two *Rental* objects are equal if they have the same *title* ignoring the case.

- **In the *Rental* class:**

➤ The overridden *compareTo* method should return the difference between the numbers of times the two compared items are checked out.

- **In the *Movie* and *VideoGame* classes:**

➤ The *computeOverDueCharge* method should find out the number of days since the movie/video game was checked out, if it is less than the maximum number of renting days, then it should return 0, otherwise, the over due charge is number of late days by the *overDuePenalty* for the movies and number of late days by twice the *overDuePenalty* for the video games.

➤ The *updateRatingAverage* method should first check that the passed rating is within the range. If not, it should throw an appropriate exception with the following message “*Rating was not accepted. Movie rating should be between 1-5*” for movies and the following message “*Rating was not accepted. Video game rating should be between 1-10*” for video games. If the rating is within the range it should increment the *numRatings* field and recompute the *ratingAverage*.

- In the *RentingStore* class:

- The *removeItem* method should remove the item with the passed *title*, it should return true if the item is found and removed successfully and false otherwise.
- The *checkOut* method should display an input dialog to the user with the following message: "*Enter the title of the item to check out*", then if the item is found and is not checked out it should change its *checkedOut* attribute to true, increment the *numTimesCheckedOut* attribute and return true, otherwise it should return false.
- The *checkIn* method should display an input dialog to the user with the following message "*Enter the title of the item to check in*", then the following message "*Enter the item rating, 0 if you do not want to rate, for movies from 1-5, for video games 1-10*". Then it should find the item with the entered title and change its *checkedOut* attribute to false then update its *ratingAverage* with the entered rating and return true, otherwise it should return false. Note that your program should check for any exceptions that could be thrown when updating the *ratingAverage* (because the rating is not within the accepted range) and handle it by displaying a message dialog that contains the exception message.

Good Luck 😊