**CMPE 202 Code Refactoring**

Student Name: Pranathi MK

Student Id: 016048214

1. **Identify a code smell in the following Java code (2 points)**

The Java code contains a code smell related to duplicate code, where certain attributes such as bookTitle/movieTitle, rentDate, dueDate, and rentalFee, along with methods like isOverdue() and lateFee(), have been repeated in multiple places. This redundancy in the code may cause maintenance issues and could be improved by refactoring.

1. **Explain how you will refactor the code to get rid of the code smell (4 points)**

* Duplicated attributes and methods can be extracted out and abstracted into a class Rental. Both BookRental and MovieRental can extend Rental class.
* Since bookTitle and movieTitle signify same meaning, it can be renamed to a generic attribute title. rentDate and dueDate, rentalFee can be moved to parent Rental class.
* However, attributes author and classification are retained in its own classes as they are not duplicates.
* Similarly, rentalFee() method is also duplicated, can be moved to parent Rental class. But lateFee() method is not exactly same but similar apart from late fee factor(0.25/0.5).
* An attribute lateFeeFactor can be introduced whose value is different in both classes. Which can be initialized upon object creation or can be initialized with a default value in its own sub classes. Alternatively, lateFee() method can also be overridden in subclasses.

1. **Write the re-factored code (14 points)**

Rental.java

public class Rental {  
  
 private String title;  
 private Date rentDate;  
 private Date dueDate;  
 private double rentalFee;  
 private double lateFeeFactor;  
  
 Rental(double lateFeeFactor) {  
 this.lateFeeFactor = lateFeeFactor;  
 }  
  
 private boolean isOverdue() {  
 Date now = new Date();  
 return dueDate.before(now);  
 }  
  
 public double lateFee() {  
 if (isOverdue()) { return rentalFee \* lateFeeFactor;}  
 else return 0;  
 }  
  
}

BookRental.java

public class BookRental extends Rental {  
  
 private String author;  
  
 BookRental() {  
 super(0.25);  
 }  
  
}

MovieRental.java

public class MovieRental extends Rental {  
  
 private int classification;  
  
 MovieRental() {  
 super(0.5);  
 }  
  
}