

**King Fahd University of Petroleum & Minerals**  
**College of Computer Sciences and Engineering**  
**Information and Computer Science Department**  
**ICS 108: Object Oriented Programming (3-3-4)**

Assignment 3 – Spring Semester 222

Deadline: Tuesday, March 14, 2023

**Instructions:**

- Please submit only one java file.
- DO NOT compress your file.

**Late submission policy:**

- Submissions received within 24 hours after the deadline will be graded out of 60% of the overall grade.
- No submissions will be accepted after 24 hours of the original deadline.

A bank offers two kinds of credit cards, silver and golden cards. With silver cards, customers pay a fee, e.g. 15%, for all card transactions. With golden cards, they pay a fixed monthly fee, e.g. 100.

**Design** the class Card that has two instance variables card no. (int) and transactions (ArrayList of doubles). It has only one constructor, with one argument, card no, that creates a card object. Include getters/ setters for all variables. Override the toString method to return card no and the total amount, including the fees, that must be paid by the card owner. Include the method **purchase(double amount)** that adds a new transaction to the transactions lists. Override the equals method. Two cards are equal if they have the same card no. The class implements Comparable, the comparison is based on the total amount, including the fees, that must be paid by the card owner.

**The subclass** SilverCard has one instance variable transactionFee (double). Include one constructor with all the arguments. Do not include getters/ setters. ~~Override the toString method to return the fee, card no, and the total amount.~~ Override the purchase method to **also** compute the fee and append it as a new transaction to the transactions list.

**The subclass** GoldenCard has one instance variable monthlyFee (int). Include one constructor with all the arguments. Do not include getters/ setters. ~~Override the toString method to return the monthly fee, card no, and the total amount.~~ Do not override the purchase method. **Do not add the monthlyFee to the transactions list.**

**Design a test** class that creates an array of type Card. Fill the array with objects from various subclasses. Do some transactions with all cards, call the method purchase with random amounts. Sort the array using Arrays.sort method based on the total amount then print it. Finally find the total amount of all GoldenCard objects in the array.

**Notes:**

- All variables are private
- You may add **only** methods, other than mentioned above, to the classes if needed