

Spring 2026 COP 3330 Program #5: Inheritance

In this program, you'll create four classes total, three of which will store the blueprint for actual objects in an inheritance chain and one to test those three classes. Here is a list of the three classes in an inheritance chain to create:

Classes to Create

Card

Instance variables: first name, last name, year (this is the year you first got your card, similar to the American Express commercials), current year

```
// Creates a new Card for a person with the name fName, lName, setting both
// the year the card was received and the current year to year.
Card (String fName, String lName, int year)

// Returns a string representation of the object.
String toString()

// Advances this Card by one year.
void advanceYear()
```

CreditCard (inherits from Card)

New instance variables: balance (double, starts at 0), credit limit (double, default is 3000), interest rate (double, store as an annual percentage, starts at 14% or 14.0)

```
// Creates a new CreditCard for a person with the name fName, lName,
// setting both the year the card was received and the current year to year,
// as well as the default credit limit and interest rate above.
CreditCard (String fName, String lName, int year)

// Returns a string representation of the object.
String toString()

// Attempts to charge amount to the card. If successful, amount is added
// to the balance of the card and true is returned. Otherwise, false is
// returned and no charge is made. A charge is only made if doing so doesn't
// make the balance greater than the credit limit.
boolean charge(double amount)

// Returns the amount paid to reduce the balance. If amount is negative,
// 0 is returned and no action is taken. If amount is greater than balance,
// then the balance is paid off in full and this value is returned.
// Otherwise, the balance is reduced by amount and amount is returned.
double payoff(double amount)

// Advances this Card by one year, which results in an increase of the
// credit limit by 1000 dollars as well.
void advanceYear()

// Adds interest to the balance of this CreditCard for one month.
// Divide the annual interest rate by 12 to get the monthly interest rate.
void addInterest()
```

GoldCreditCard (inherits from CreditCard)

New instance variables: frequent flyer miles (int, starts at 0), (also starts at a default credit limit of 7500 and interest rate of 12%)

```
// Creates a new GoldCreditCard for a person with the name fName, lName,  
// setting both the year the card was received and the current year to year,  
// as well all of the defaults mentioned above and in CreditCad.  
GoldCreditCard (String fName, String lName, int year)  
  
// Returns a string representation of the object.  
String toString()  
  
// Attempts to charge amount to the card. If successful, amount is added  
// to the balance of the card and true is returned. Otherwise, false is  
// returned and no charge is made. A charge is only made if doing so doesn't  
// make the balance greater than the credit limit. Also adds 2 frequent  
// flyer miles for each full dollar spent.  
boolean charge(double amount)  
  
// Advances this Card by one year, which results in an increase of the  
// credit limit by 2000 dollars and adds 10000 frequent flyer miles.  
void advanceYear()
```

Testing Class to Create

TestCredit

This class will have your main. Feel free to break this class up into several different static methods. You will present the user with the following menu:

- 1) Create a credit card
- 2) Make a purchase
- 3) Pay a credit card bill
- 4) Advance to the next year
- 5) Print out the status of all credit cards
- 6) Quit

Your main will have an array of size 10 of Card. For the purposes of this assignment assume the user will never try to create more than 10 Credit Card objects.

When a user chooses option (1), you will ask them if they want a regular card or a gold card. After they answer, ask them for all of their information and create their card for them. Tell them the number of their card (it will be 0 through 9, the first one created will be 0, then 1, then 2, etc.)

When a user chooses option (2), ask them which card they want to use (0 through 9). Then ask them how much their purchase was. If possible, execute the purchase with that card. If not, give them an error message saying that the purchase would exceed their credit limit and was not made.

When a user chooses option (3), ask them for which card (0 through 9). Then show them their current balance and ask how much they are paying. Adjust their account accordingly. Then, after the payment, go ahead and add interest to the card. (If the card was paid down to 0, then no interest will be added when this method is called.)

When the user chooses option (4), advance all cards by one year.

When the user chooses option (5), simply list all the information for each card in order (0 through 9).

If time permits, feel free to add another class, PlatinumCard, which is even nicer than the GoldCard.

Please try to do some error checking. For example, try to prevent negative charges or payments. Also, try to prevent a person from using a card that doesn't exist. (For example, if they have created two cards, numbered 0 and 1, but when asked which card they want to use, they choose 5, you should simply say that card doesn't exist.)

I am leaving this specification open to some interpretation. The idea is to make this a plausible simulation and to have the inheritance specified by the three classes.

Deliverables

A minimum of four classes stored in four files:

Card.java

CreditCard.java

GoldCreditCard.java

TestCredit.java

If you want to add a fifth class which inherits from either CreditCard OR GoldCreditCard, feel free to do so.

Make sure to include a header comment for each file you submit and appropriate internal comments.