# ITEC 3150, Homework #1 - Intermediate Programming Review

### Due Date

The due date is published in banner. Note that any weekend dates are provided as a convenience to you. I may not answer questions until business hours on the following Monday. NO LATE ASSIGNMENTS WILL BE ACCEPTED. This is a 3000-level class - you will need to manage you time effectively.

### Coding Standards Note

The coding standards are in a document in course OneDrive folder. Recall that I want to see code that has good naming for classes, variables and methods. Your code should be well organized and very readable. Areas of ambiguity should be commented, but comments may need be sparse beyond that. You do not need to add Javadoc style commenting.

### Submission

* Submit your zipped project file (IntelliJ)
* Provide an expository video describing your solution. Recall this video will help with creation of your video. The video must be uploaded or shared privately. Missing or incorrect videos will result in a grade of zero.

### Competencies being graded

1. Ability to read from a text file
2. Ability to use inheritance to create a set of subclasses all inheriting from a common parent class.
3. Using polymorphism to create and maintain a single list of the subclasses
4. Use System.out.println and Scanner to create a user friendly interface into your program
5. Ability to read an English language problem description and design a multi class solution in Java
6. Ability to follow given coding standards- in D2L content under Coding Standards.

### Problem Statement

Write a Java program that simulates a simple banking teller machine. The machine contains a list of account holders and their balance. There are two types of accounts- checking and credit card. Both accounts contain a unique account number, a name, a type (“checking” or “credit card”) and a balance. In a checking account, the balance is the amount of money available for withdrawal. In a credit card, the balance is the total amount withdrawn. The credit card account also contains a credit limit which is the maximum amount the account holder can withdraw.

Upon starting the program, the list of accounts will be read from a text file named accounts.txt. A text file containing information on 5 accounts is provided with this assignment.

The program will provide an ongoing loop that prompts for account number and amount of cash to receive. The account number must match an existing account- if the number is invalid, the user will be notified and the transaction ended.

If the account is a checking account, the amount of cash requested will be checked to make sure there is enough in the balance to cover the cash withdrawal. If there is, the cash amount will be deducted from the account balance and the new balance printed to the user.

If the account is a credit card account, the amount of cash requested will be added to the balance if that addition will not cause the balance to exceed the credit limit. If it does exceed, the transaction will be canceled and the balance left unchanged. If not, the amount withdrawn will be added to the balance and the new balance printed to the user.

An example of user interface: (user inputs are in italics)

Please enter an account number: *1234*

I’m sorry I cannot find that account - please try again.

Please enter an account number: *2222*

Please enter amount of desired cash: *100*

This is a checking account, pick up cash in cash slot. Your new balance is $300.

Do you want another transaction (Yes/No)? *Y*

Please enter an account number: *2223*

Please enter amount of desired cash: *100*

This is a credit card account, pick up cash in cash slot, your new balance is $600.

Do you want another transaction (Yes/No)? *Y*

Please enter an account number: *2223*

Please enter amount of desired cash: *100*

I’m sorry this will exceed your credit limit. Please try another account.

Do you want another transaction (Yes/No)? *N*

### Rubric

|  |  |
| --- | --- |
| Compiles and has minimum of 3 separate Java classes and files: | 10 |
| Correctly uses inheritance for account types | 20 |
| Correctly uses exception handling (no throws) | 15 |
| Correctly removes withdrawal from checking acct balance | 10 |
| Correctly adds withdrawal to credit card balance | 10 |
| Correctly reads text file | 10 |
| Correctly writes text file contents to the console at end of session | 10 |
| Meets coding standards | 15 |