

# Bank on it (A7)

3/10/2023

## 74/100 Points

Attempt 1



Review Feedback  
3/10/2023

Attempt 1 Score:  
74/100



View Feedback

Anonymous Grading: no

### Unlimited Attempts Allowed

#### Details

## Overview

You have been contacted by officials at RETH (Ridiculously Easy to Hack) bank. They would like you to create a basic ATM system for them. Here are the requirements:

- Each user will have an account number and a PIN
- Each user will have a checking account and a savings account.
- The checking account has a balance, and the user can deposit money into the account, withdraw money from the account, and check the balance.
- The user cannot withdraw more money from the account than the current balance
- The checking account DOES NOT have interest.
- The savings account is almost exactly like the checking account EXCEPT:
  - There is an interest rate associated with the account
  - The interest can be calculated for a certain period
  - Interest is calculated by the standard basic interest formula:
    - $A = P(1 + rt)$
    - P = principle (balance before calculating interest)
    - r is the interest rate
    - t is the period (months)
- The ATM will expect a user to log in using account number and PIN
- If the user is an administrator, she will be taken to an admin menu
- If the user is a known user, she will be taken to a user menu
- If the user is unknown, the system will exit.
- The admin user will have an account number of 00000 and a pin 12345
- The admin menu will have the following options:
  - add a user
  - delete a user (account must be empty)
  - list all users
  - apply interest to all savings accounts
  - exit
- The user menu will have the following options:
  - get checking balance
  - deposit into checking
  - withdraw from checking
  - get savings balance
  - deposit into savings

- withdraw from savings
- exit
- The system will need to store the user and account information into some sort of file system.
- In keeping with our bank motto: "efficiency through complete disregard for user security" you do not need to implement any sort of security mechanism on the file information.

## Technical Considerations

- This is an Object-oriented project. You will definitely need to break the problem into objects
- Typical solutions use between four and six classes
- No UML is provided. It is your first job to generate a plausible object scheme.
- You will definitely need a UML diagram, and you should update it regularly to reflect the reality of your code. Don't wait for the last minute to do your UML.
- At least one class should inherit from another class in your system (not Object- that's a given.) Look over the description again to see if you can figure out where inheritance will be useful.
- Write the program in Java (you can do a version in C++ for blackbelt if you want.)
- Include a makefile with clean, run, and debug targets (valgrind is not necessary for Java programs)
- You do NOT need a custom package for this assignment. The default package is fine. If you use a custom package you need to ensure the program still runs correctly with a 'make run' from the main directory.
- Consider input integrity. You might need some exception handlers to ensure input is correctly formatted.
- Utilize appropriate exception handling around file I/O.
- You MAY use prepackaged containers such as Vector or ArrayList for your data management.
- You MAY also use object serialization to store and retrieve your data

## Algorithm

As always, begin with an algorithm in markdown(md) format. This document should begin by describing the main purpose of the program using the Goals - Input - Output - Steps technique we've been using. For EACH function/method you have in the program or classes, you must go through the same process (Defining goals, input, output, steps). Remember that your algorithm should be complete enough that by the time you are ready to start writing code, you pretty much know what you are going to do. You should get mostly through the algorithm step together in the recitation session, but definitely try to write this on your own as well.

Your algorithm file must be in a .md file named: algorithm.md

Points will be taken off if it is not named this exactly, or if it is in another file type (including but not limited to; .rtf, .docx, .doc, .pdf, ... etc.)

## UML Diagram

You will need to turn in a UML diagram displaying all classes used in your program. The UML diagram should be in a standard image file. (png, svg, gif)

Include the file in your github repo under the base folder, and if you have more classes for your blackbelt, you should make another one that reflects these changes.

## Turning in the project

- This assignment will be turned in through iu github. Please name your repo CSCI24000\_spring23\_A7
- If your repo has another name, it will not be graded.
- After you've submitted your assignment in Github, come back to this assignment page and submit the full Github URL (<https://github.iu.edu/username/reponame> → <https://github.iu.edu/username/reponame>) for your repo here.
- Your repo must be private. We will not grade projects in public repos, and we will require you to take down any homework assignments in a public repo.
- Please ensure to add all the following are listed as collaborators:
- **Git Collaborators** (<https://iu.instructure.com/courses/2131288/pages/assignments-submission-and-grading>)

Note that your code will be tested with an online plagiarism tool. Please DO NOT turn in work that is not yours. We will know, and we will be displeased.

## BlackBelt extension:

This is a wide open assignment, so blackbelt extension options abound:

- Modify the system to allow each user to have any number of accounts
- Re-write the system in C++ using the same UML
- Look into abstraction and see how that might improve this project - with an interface and / or abstract class
- Add a security mechanism to the data - look into hashing algorithms for this.
- Whatever else you think could enhance this project

As always, use github to submit your project.

[https://github.iu.edu/parmsing/CSCI24000\\_spring23\\_A7](https://github.iu.edu/parmsing/CSCI24000_spring23_A7)



<https://iu.instructure.com/courses/2131288/modules/items/28790628>



<https://iu.instructure.com/courses/2131288/modules>

*You are unable to submit to this assignment as your enrollment in this course has been concluded.*