COSC1408 Foundations of Computer Science II
Term Project
Due Date: Sunday, 5/5/2017

This is a group project. You can have two students in a group.

Problem description: See page 969 group project: Bank Accounts

- If withdraw is greater than the current balance, a NotEnoughFund exception is thrown.
- In addition to the three classes (base, checking, and savings) for the bank account, you need to define classes for clients who are different types of Employees:

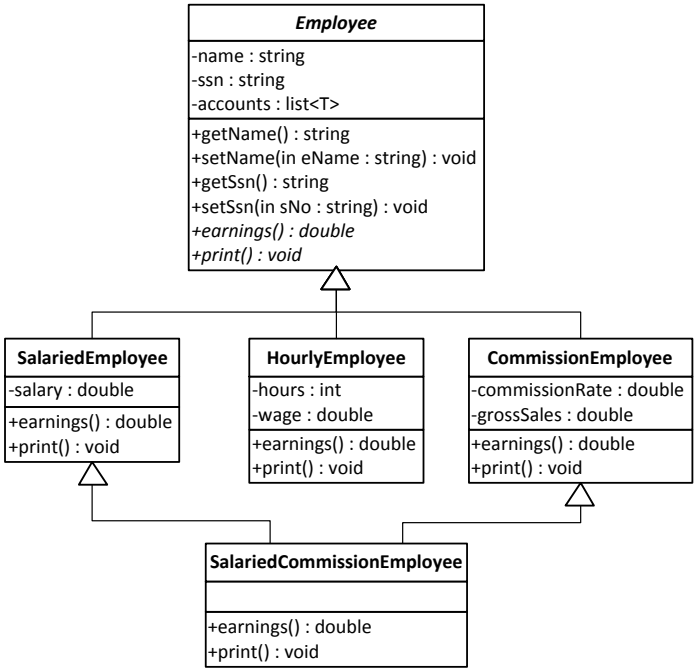
  SalariedEmployee: earnings = weekly salary

  HourlyEmployee: if hours <= 40 earnings = wage * hours;
  if hours > 40 earnings = (40*wage) + ((hours – 40) * wage * 1.5)

  CommissionEmploye: earnings = commissionRate * grossSales

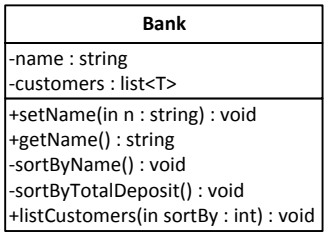
  SalariedCommissionEmployee: earnings = weeklySalary + (commissionRate * grossSale)

- Each client uses a list data structure to store his/her bank accounts. You can define your own linked list or use the `list` container in STL.



- addAccount(BankAccount *)
- The `print` function prints Employee's name, ssn, salary, and bank accounts information.
- The `earnings` function returns employee's salary.

- Define a `Bank` class. The bank uses an `array` or a `list` to store the customers (you can use the `list` container in STL).



- The `listCustomers` function lists all customers and their accounts information.
- To test your program:
  - Create at least one employee for each type of employees.
  - Each employee has at least one checking account and one savings account.
  - Design your testing program to test all functions defined for bank, bank accounts and employees.

Project levels

Basic: 100 points

- Employee classes:
  - Employee, SalariedEmployee, HourlyEmployee, CommissionEmployee
- Account classes:
  - BankAccount
    - Attribute: balance
    - Functions: default constructor, constructor accepts an argument for the balance, deposit, withdraw.
    - deposit(double): balance = balance + deposit
    - withdraw(double): can't withdraw if after-balance is negative, otherwise balance = balance - withdraw.
  - CheckingAccount
    - Functions: default constructor, constructor accepts an argument for the balance, override deposit and withdraw defined in BankAccount.
    - deposit(double): if balance is lower than $50.00 the deposit has to be at least $20.
    - withdraw(double): can't withdraw if after-balance is negative, or $10 penalty if after-balance is lower than $50.
  - SavingsAccount
    - Functions: default constructor, constructor accepts an argument for the balance, override deposit and withdraw defined in BankAccount.
    - deposit(double): if balance is lower than $100.00 the deposit has to be at least $50.
    - withdraw(double): can't withdraw if after-balance is negative, or $30 penalty if after-balance is lower than $100.

Bonus: 110 points

- All employee classes described above and all account classes described in the group project on page 969.