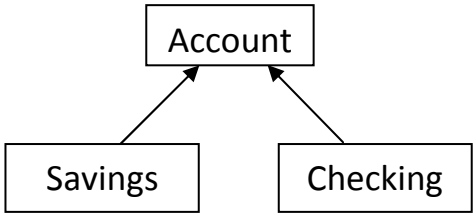
Unit 5 – WS04 Banking Objects

Name _____

The goal of this assignment is to expand the banking classes discussed in a classroom to implement the following inheritance hierarchy:



Problem #1 – Create the two classes (Account and Savings) that were discussed in a classroom. The code for these classes is on page 2 of this worksheet.

Problem #2 – Create the accountMain class and code the accountMain class to test all aspects of the Account class and the Savings class.

Problem #3 – Add one more class called Checking that has an instance variable balance that tracks the balance of checking accounts. It should initialize the balance variable to zero when a checking account is created.

Problem #4 – Edit the Checking class so that it also contains an ArrayList called checks. When a checking account is created, the initial value of the checks ArrayList should be set to 0.0 (index 0 = 0.0). This means that every check thereafter will correspond to the index number if the checks are entered in order.

Problem #5 – Give the Checking class a mutator method called deposit( ) that takes in a deposit amount as a parameter and adds the amount to the balance.

Problem #6 – Give the Checking class a mutator method called withdraw( ) that takes in a withdraw amount and subtracts the amount from the account balance.

Problem #7 – Add another mutator method called earnInterest( ) that gives the account balance 0.1% interest when called.

Problem #8 – Add another mutator method called enterAccountInfo( ) that has four parameters. An integer account number, a String for the account password, a String for the username, and a double for the initial balance in the account. This method should store all of these passed parameter values into a checking account object.

Problem #9 – Create an accessor method called getAccountNumber( ) that overrides the Account class getAccountNumber( ) method (this presents and example of polymorphism). Checking account numbers should be 9 digits and always start with 888.

Problem #10 – Create a mutator method name writeCheck( ). This method should take in a double parameter that represents the amount the check was written for. The writeCheck( ) method should then add the check amount to the checks ArrayList and subtract the check amount from the account balance.

Problem #11 – Create one last accessor method named checkLookUp( ). This method should be passed an integer check number so that it can look up the check amount by finding the index number that corresponds to the check number. The method should return how much the check was written for.

Problem #12 – Edit your accountMain class to include code that tests and verifies all aspects of all classes. Your accountMain class should show that all classes and all method are working correctly … like you almost have a working bank!

```
public class Account {
   int number;
   String name;
   String password;
   public Account(){
      number=0;
      name=" ";
      password="XYZ123";
   }
   public void enterAccountInfo(int num, String nm, String pw){
      this.number = num;
      this.name=nm;
      this.password=pw;
   }
   public int getAccountNumber(){
      return this.number;
   }
   public String getPassword(){
      return this.password;
   }
   public void changePassword(){
      Scanner getPW = new Scanner(System.in);
      System.out.println("Enter your new password: ");
      String newPW = getPW.next();
      System.out.println("Re-enter your new password: ");
      String verifyPW = getPW.next();
      if(newPW.equals(verifyPW))
         this.password=newPW;
      else
         System.out.println("Sorry, no password change!");
   }
   public String getName(){
      return this.name;
   }
}
```

```
public class Savings extends Account{
   double balance;
   public Savings(){
      balance=0;
   }
   public void deposit(double dep){
      this.balance+=dep;
   }
   public void withdraw(double wd){
      this.balance-=wd;
   }
   //no override needed, this is method overloading- diff parameters!
   public void enterAccountInfo(int num, String nm, String pw, double
bal){
      this.number=num;
      this.name=nm;
      this.password=pw;
      this.balance=bal;
   }
   public void earnInterest(){
      this.balance=(this.balance*1.005);
   }
   @Override                    //this is override, replaces!
   public int getAccountNumber(){
      return this.number+888000000;      //start w/888 & 9 digits
   }
}
```