

Chapter 11 – Inheritance and ArrayLists -- helper file

See the file **Chapter11HelperFile.zip** for sample solution files.

1. Download the attached file (**Account.java**) in the same folder that you plan to use for your program for the following question. As you can see from the class file, the class Account has member attributes for account number (id), balance, annual interest rate and date created, and methods to deposit and withdraw funds. Create 2 subclasses for Checking and Savings accounts. A checking account has an overdraft limit of \$10,000. So, you need a new attribute for this class. The savings account has an overdraft limit of \$2000 and a monthly fee of \$5. Override the toString methods of each of the subclasses to specify a note about the type of the account (Checking or saving). Write a test class that creates objects from the Account, SavingsAccount and CheckingAccount, assign values and exercise the various methods of the classes. **Make sure to exercise ALL of the methods as you test your code. A minimal testing will not gain many points.** If I counted correctly, you will have submit 4 .java files (the Account.java that I provided, the 2 subclasses and the main testing class).
2. (**For sample solution, see ArrayListUsage.java included in the ZIP file**). This can be done in one file. There is no need to create a utility class. Write a program that uses the ArrayList (covered in 11.11 – page 432-chapter 11) class to store objects. Do NOT use static arrays that we used in earlier chapters. Listing 11.8 on page 433 may be helpful. Write a program to store the name of employees, their pay rate and hours worked using 3 ArrayLists (one to hold the name, one to hold the pay rate and one to hold the hours worked)
Once you have the data in the ArrayLists, write code to:
 - a) Create a new array list to hold the employee pay by multiplying the rate by hours.
 - b) Display the size of the array lists
 - c) Ask the user to enter the name of an employee and check if the name exists in the array list. The input name can be entered in any case, so your search must be case insensitive. Display the message “EMPLOYEE found”, if the name is found in the list. Otherwise, display “EMPLOYEE NOT found”. EMPLOYEE is the name of the employee entered by the user.
 - d) Allow the user to add more names.
 - e) Allow the user to remove names. If the name to be removed cannot be found, display an error message.
 - f) Display the contents of the array lists using its toString method.
 - g) Allow the user to enter an employee name and display the corresponding pay. If the input name does not exist, display an error message.