Lab 3 Objectives

1. Example of creating a comparator

For your first analytic response that is to show the

* **average income for males vs. females**

you may want to create a comparator class to sort by sex

For example…

**import** java.util.Comparator;

**public** **class** SexComparator **implements** Comparator<BankRecords>{

@Override

**public** **int** compare(BankRecords o1, BankRecords o2) {

// use compareTo to compare strings

**int** result = o1.getSex().compareTo(o2.getSex());

**return** result;

}

}

\*Think of comparators you can set up with not only a primary sort but secondary sorts as well. Example maybe a female sort as a primary sort and mortgages as a secondary sort, etc.

Another example here maybe to create a comparator java class for sorting by location…

**import** java.util.Comparator;

**public** **class** LocationComparator **implements** Comparator<BankRecords>{

@Override

**public** **int** compare(BankRecords o1, BankRecords o2) {

// use compareTo to compare strings

**int** result = o1.getRegion().compareTo(o2.getRegion());

**return** result;

}

}

2. Call up your comparators in your Records.java file.

Ex. in main(), you can call a series of functions to perform your analytic results.

Set up your file to also not only write to the console but to a text file.

See example that follows…

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.util.Arrays;

**public** **class** Records **extends** BankRecords {

//create formatted object to write output directly to console & file

**static** FileWriter *fw* = **null**;

**public** Records() {

**try** {

*fw* = **new** FileWriter("bankrecords.txt");

} **catch** (IOException e) {

e.printStackTrace();

}

}

**public** **static** **void** main(String[] args) {

Records br = **new** Records();

br.readData();

// call functions to perform analytics

*AvgComp*(); // analyze average income per gender

//femsComp(); // female count w. mort/savings accounts

//malesComp(); // male counts per loc. w. car & 1 child

// \*\*\* close out file object \*\*\*//

**try** {

*fw*.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}

**private** **static** **void** AvgComp() {

Arrays.*sort*(*robjs*, **new** SexComparator());

// set up needed variables to gather counts & income by sex

// to determine average income by sex

**int** maleCt = 0, femCt = 0;

**double** maleInc =0, femInc = 0;

**for** (**int** i = 0; i < *robjs*.length; i++)

**if** (*robjs*[i].getSex().equals("FEMALE")) {

++femCt;

femInc += *robjs*[i].getIncome();

}

**else** {

//...

}

// display resulting averages to console and to file

System.***out***.printf(" ... ");

**try** {

*fw*.write("Avg inc. for Females: $" + (femInc/femCt) );

*fw*.write("...");

} **catch** (IOException e) {

e.printStackTrace();

}

}

}

Continue in like manner as above to finish up two other method definitions,

callable in main().