The University of Mississippi
CSCI 112: Project 5
Spring 2023

Due Date: Thursday April 27 by 11:59 pm

*** Late programs will have points deducted each day. If your program does not compile, it will not receive any credit. If you need help, come see me, the class TA, or a department tutor during our office hours.**

This project contains aggregation "has a" relationships only, there is no inheritance between classes.

The Oxford University Art Council (OUAC – is a fictious organization for this project only) is having a silent auction to raise money to help local artist's and galleries. The prizes for the auction have been donated by various companies and individuals in and around the Oxford Community. You will write a program that reads data from 2 text files and arranges the data according to the specifications below. In this graded project, you are tasked with creating a Java project representing the following problem using the classes below.

The class's needed are:

- Bidder class that has class attributes of: a Bidders name, a hometown, and a bid amount for the specific auction item. This is a normal regular object that you should know how to create by now with the 4 normal types of methods.

- AuctionItem class that has class attributes of: an item description, an item number, a Bidder's Arraylist or array, and the items total bid amount. You will also need to keep track of the number of bids made on the item, the items highest bid, the total amount bid on the item by all the Bidder's, and the item's average bid amount (these may or may not be attributes, but simply calculated or obtained).

- OUAC class that has class attributes of: the date, a AuctionItem ArrayList or array, and the total amount raised for the auction from the winning (highest) bids on all auction items.

- P5Driver class that has a OUAC Object and reads through the text files.

- Junit Test classes should test all the methods (excluding print and write to text file methods) of the Bidder, AuctionItem, and OUAC classes.

You Program should:

- Have introduction and closing messages

- Only instantiate a OUAC object (none of the other class objects) in the Driver.

- Read the data from the “AuctionItem.txt” and “Bidders.txt” files in the Driver and pass the data to the appropriate methods.

- Calculate the total bid amounts for each auction item as we as the total raised by the auction.

- Print all the data (attributes) for the OUAC including the AuctionItem and Bidder’s (including how many of each), highest bid, average bid, and totals.

- Sort each AuctionItem’s Bidder’s data by each Bidder’s bid amount in descending order.

- Inform the user the data is sorted and Re-Print all the data (all the data from the previous print) for the OUAC (AuctionItem, Bidders, etc..)

- Print award winners (you can come up with the award names) for the item with the highest bid (the amount and bidder), the item that had the most bids, and the item with the fewest bids.

- Write all the data (the same info from the Re-Print all the data section after the sorting) for the OUAC (AuctionItem, Bidders, totals, etc..), and the awards information to a text file (you can name it whatever you want, but do not write to the original files you read from). The data in the text file should be as legible as the output you printed to the monitor.

Specific requirements/information:

- The object classes should all have Constructors, setters, and getters according to their attributes, a toString or print method, and add/calc methods as needed.

- The sort should be one of the recursive sorts discussed in class (either quick or merge).

The Text Files:

- The “AuctionItem.txt” file per line contains: an item description and an item number.
- Each line of the “Bidders.txt” file per line contains a person’s name, hometown, the number of the item they are bidding on, and the bid amount. Each person bid on multiple items.

* **Note::** The writing to your text file could take place in a couple of classes or just one class. That is up to your implementation of the code and logic.