

Criterion C

Summary of all techniques employed

- for loop
- nested loops
- Conditional statements
- method returning a value
- user defined objects made from an OOP "template" class
- encapsulation of private methods that work on public attribute of a "template" class
- simple and compound selection (if/else)
- sorting (bubble sort etc.), and in particular sorting an array of objects based on one key attribute
- Binary Search
- Selection sort
- Saving to a csv
- Arraylist
- error handling
- Option pane generation for communicating with the user
- GUI tabs
- GUI popup menus
- Radio buttons
- Display tables
- overloaded constructors, which work differently depending on the parameters sent
- inheritance between a superclass and a subclass

Structure of program

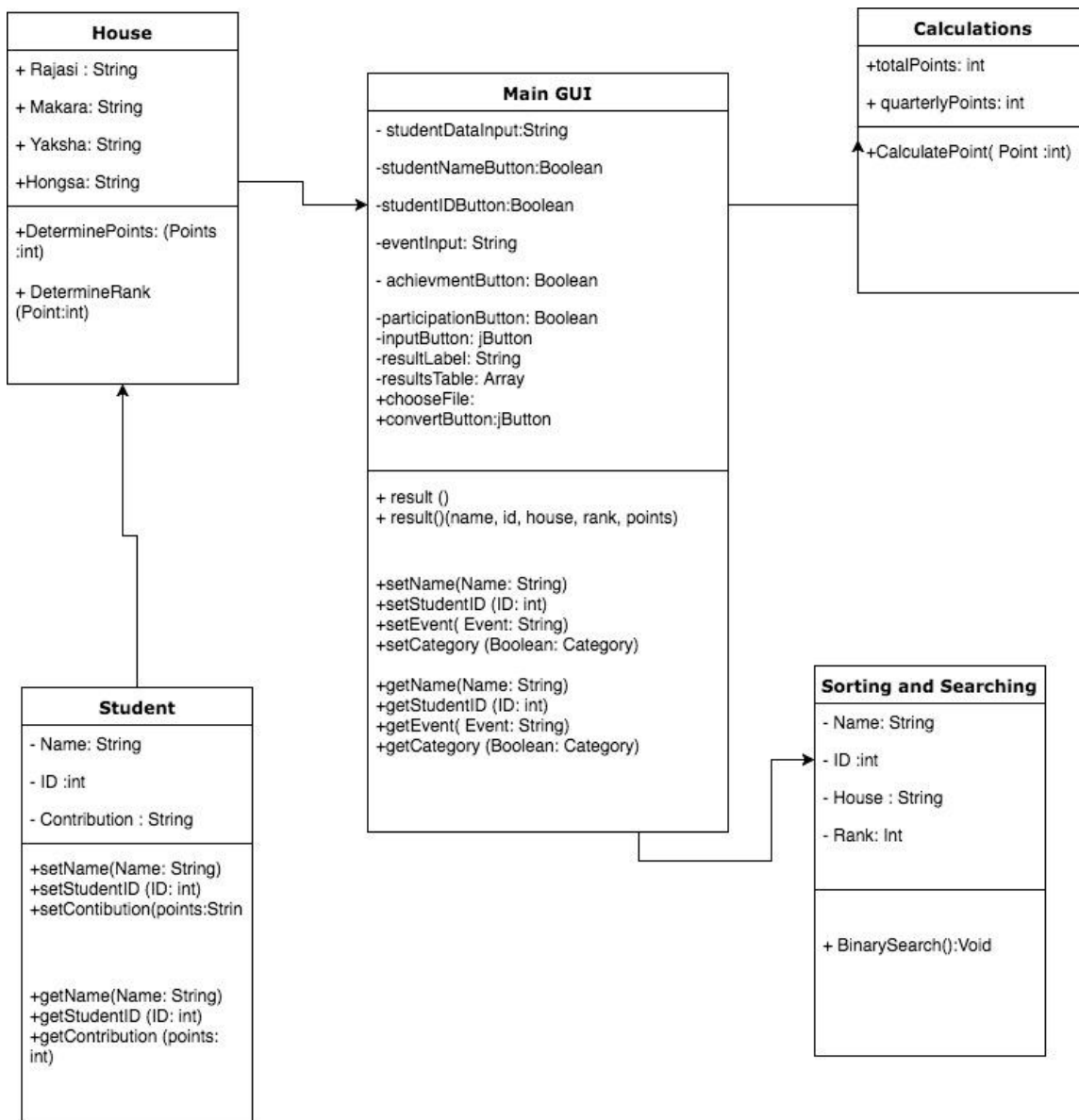
The programmed solution has been segregated into 5 distinct classes; "MainGUI", "SearchingAndSorting", "Students", "Calculations", and "House". This being said, the project is an Object oriented database.

The "MainGUI" is the backbone of my entire program and is what connects and calls upon all the other classes. The main gui also contains code for the functioning of the front-end of the program to make user interaction as seamless as possible.

The Searching and Sorting class contains two methods that will be called upon upon interaction with the search and refresh button. The two methods that will be called upon are my binary search algorithm and selection sort.

Next, my calculation class contains algorithms for allocating particular house points to a particular house or students. This class contains one method which is structured around a conditional to determine if it is an achievement point or participation point category and also includes a for loop to add up the points along with the actual algorithm for calculating house points.

The students and house class, are both template classes where my variables are defined, and where the sets and gets have been placed.



Data Structures Used

The solution is centered around two fundamental ArrayList. My student arrayList and my calculation arrayList. The student arrayList uses the template classes compiled in the student class as well as the the data inputted from the GUI pertaining to student information (Name, ID, and House the student is in). Then the calculation arrayList uses methods that were created in the calculations house points and also gets inputted data from the GUI pertaining to points(Category, and quantity of points). Finally the house class will use data from both arrayList to summarize house points gathered from each house and rank them accordingly.

```
public class MainGUI extends javax.swing.JFrame {  
    private ArrayList<Student> studentArrayList = new ArrayList<Student>(); //establisheds student array  
    private ArrayList<Calculations> calculateArrayList = new ArrayList<Calculations>(); //establisheds student array  
    studentArrayList.add(new Student(  
        studentDataInput.getText(), Integer.parseInt(studentIDTF.getText()),  
        rajasiButton.isSelected(), hongsaButton.isSelected(),  
        yakshaButton.isSelected(), makaraButton.isSelected(), eventTF.getText(),
```

GUI Components Used

The GUI components used this solutions provides a highly interactive user interface to allow our client to input and view data with ease. The program features 4 tabbed panes to divides up the program based of its four main functionalities.

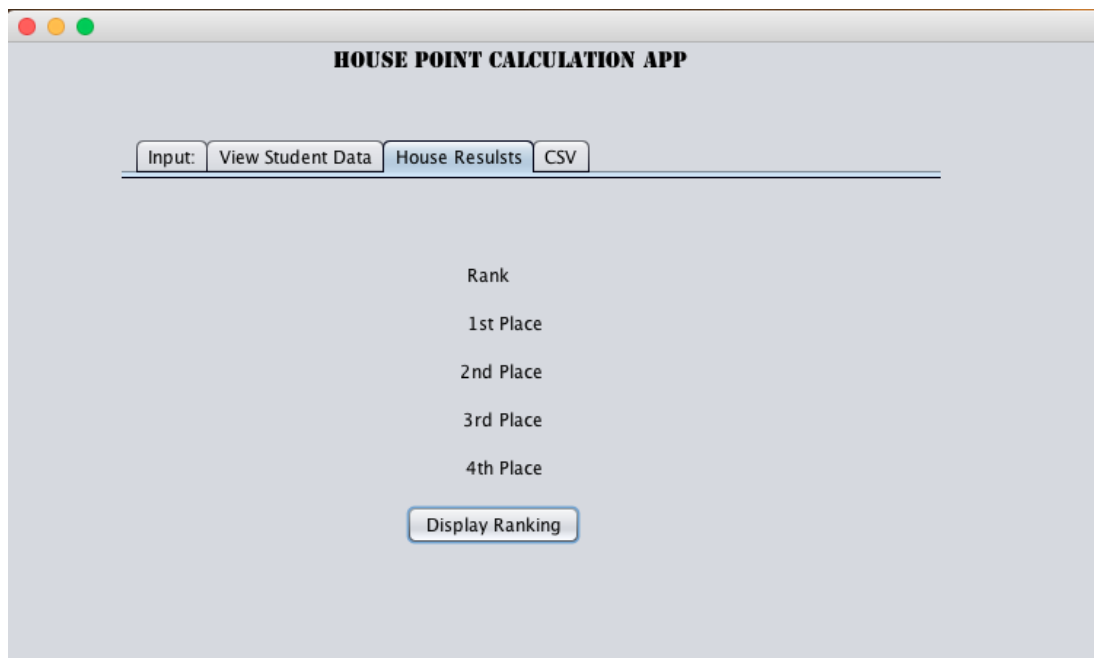
The first tabbed pain, mainly consists of JTextFields and Radio buttons in order for the user to input string data such as their student ID, name, and house event, as well as boolean data such as rank, participant range, or point category.

The screenshot displays a Java Swing window titled "HOUSE POINT CALCULATION APP". At the top, there are four tabs: "Input", "View Student Data", "House Results", and "CSV". The "Input" tab is currently selected. The form contains the following elements:

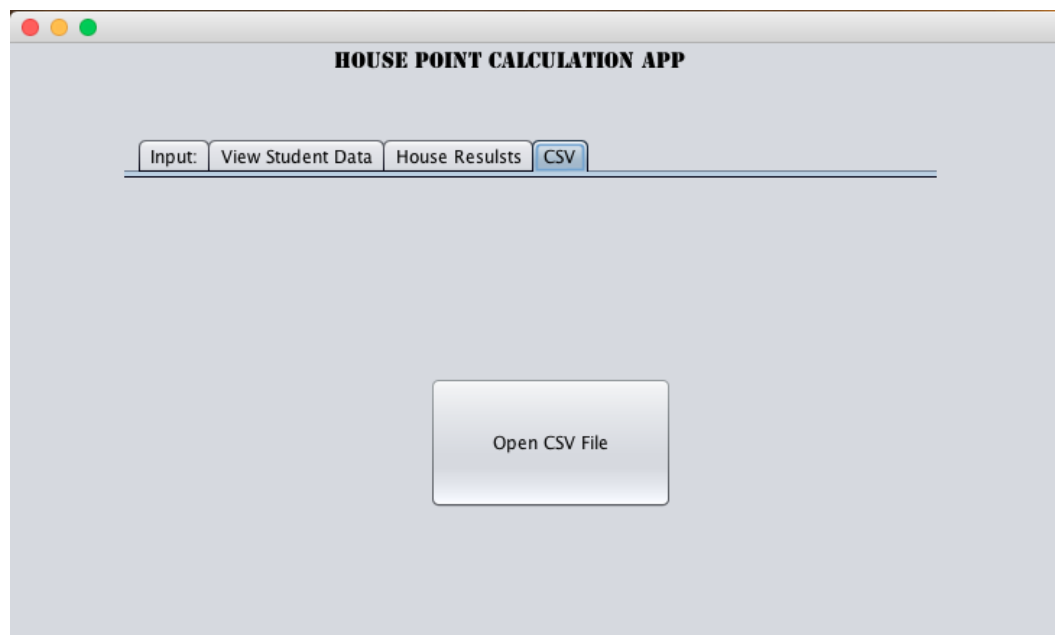
- Student Name: Student ID:
- House: Rajasi Makara Hongsa Yaskha
- Event:
- Point Category: Achievement Participation
- Rank: 0-10 11-20 21-30 30+
- Buttons:

The second, tabbed allows the user/client to view the inputted data through a display table, which has columns for, the student id, name, point category, house, and actual number of points gathered, there is also a text field which ultimately uses a binary search algorithm to search for the specific student ID that the client inputted.

Moreover, on the house rank tab presents the ranks and cumulative rank of each house to the user using JLabels.



And finally, the last tab enables the user to save the inputted data into a csv file by pressing on the JButton.



Software Tools Used

Primarily, the two software tools employed in facilitating the success of this solutions. The two being an Integrated Development Environment (IDE) and the other being a Spreadsheet software. The IDE I used was Netbeans, this software allowed me to debug, catch errors in my programs, format the code, provides suggestions of basic functions, and creates generated code for creating GUIs. I also used excel to test weather the CSV file successfully saves all the data in the correct structure and format.

