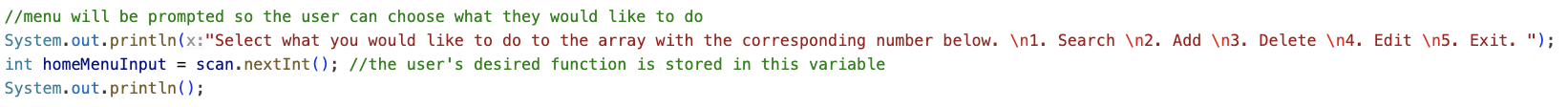
**Student Information Class**

It is the same concept with Course Information Class (different variables and header)

**Function summarization of the program:**

The program starts off with printing the defaulted students and their corresponding information (Student ID, First Name, Last Name, Birth Date, Mobile Number, Email, Gender, and Address). The user is prompted with 5 options: Search, Add, Delete, Edit, and Exit. Throughout the code, many switch statements are used instead of an if-else-if statements because the cases are used for fixed values. Whichever the user selects based on the UI menu, specific code is run. The user can keep on engaging with the program for as long as they wish until they select Exit. Once they select this function, the most updated array with the up-to-date student information is then exported to a file.





A close-up of a computer screen

Description automatically generated

**Search:**

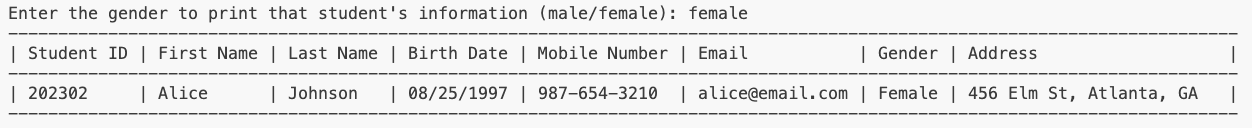
If the user selects “search,” they are then prompted with another menu of different functions. The user can search via student ID, first name, last name, birthday, mobile number, email, gender, or address. For example, if the student selects “gender,” they are prompted to enter in a gender (female/male). Their input is stored into a variable, userGender, and is then compared to each of the genders in the arrayList, using an if-else-if statement. The parameter is using the whole array, studentSampleData, and getting the gender at that specific index. Let’s say they input female, then all the student’s information would be printed if they are female. The same goes for the rest of the headers.

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



**Add:**

If the user selects “add,” then they are prompted with a series of questions to input a new student. Each of the questions line up with the header. When they answer the corresponding questions that match up with each of the necessary categories, their input is stored into a variable that is later used to print out the new student’s information. Once they finish entering all the information for the new student, the information is printed.

A computer code with many text

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

**Delete:**  
If the user selects “delete,” then all the updated student’s information is printed. The user is prompted with a question to see which student they would like to delete. Once selected, all that student’s information is deleted, and the rest of the students are displayed.

A computer code with many colorful text

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated

**Edit:**  
If the user selects “edit,” all the updated student’s information is printed, and then a question of which student to edit is prompted. Once the user selects the desired student to edit, all the questions are prompted for the user to enter the updated information for that student. After updating the information, all the student information is displayed.

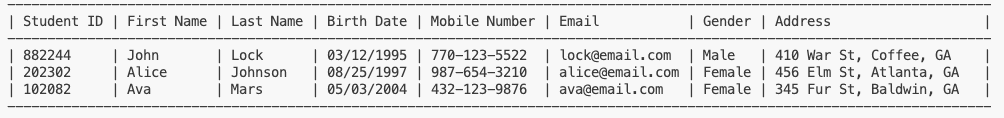
A screenshot of a computer code

Description automatically generatedA screenshot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated



**Exit:**

A computer screen shot of a program

Description automatically generatedA screenshot of a phone number

Description automatically generatedA screen shot of a computer code

Description automatically generatedThe user can search, add, delete, and edit as many times as they wish, and the arrayList will continue to update. Once the user selects “exit,” all the up-to-date student information is exported to a file, and the program ends.