DSA Week 17 activities

This week, you are required to complete two questionnaires and two labs.

- ***a.*** In this print out, answer all Week 17 questions.
- ***b.*** Also, in this print out, complete Week 17 lab 1 & 2 using the lab computers.

Note: You can complete the activities in any order, however, make an effort to complete and understand everything which prepares you for well for the Final Exam.

DSA Week 17 Questions

1. What is a Map ADT?

A map is an abstract data type designed to efficiently store and retrieve values based upon a uniquely identifying search key for each.

A map stores key-value pairs (k,v), which we call entries, where k is the key and v is its corresponding value. Keys are required to be unique, so that the association of keys to values defines a mapping.

2. Discuss the get(k), put(k, v), remove(k) and isEmpty() methods used with Map ADT.

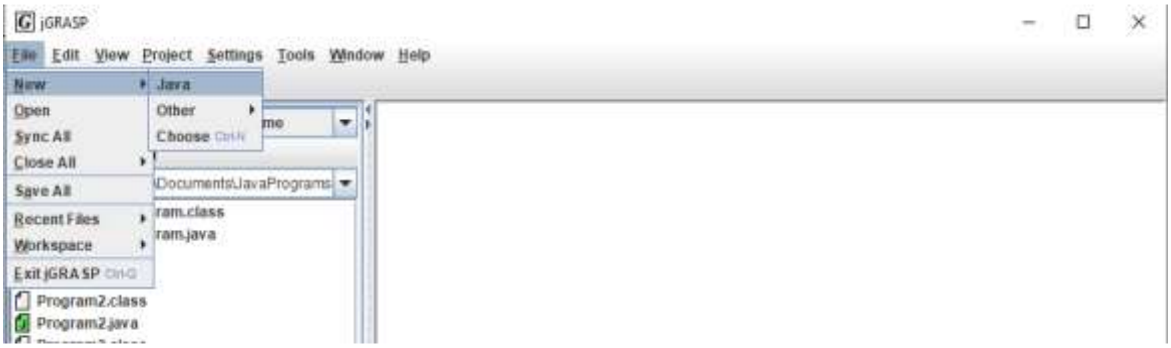
The following methods are used with the Map ADT to manipulate the data structure:

- **get(k)**: This method retrieves the value associated with the key **k**. If the key exists in the map, it returns the corresponding value; otherwise, it may return null or an equivalent indication that the key is not present.
- **put(k, v)**: Adds a new key-value pair (k, v) to the map. If the key **k** already exists, this method updates its value to **v**, replacing the previous value.
- **remove(k)**: Removes the key **k** and its associated value from the map. If the key does not exist, the method typically does nothing or returns an indication that no removal occurred.
- **isEmpty()**: Checks whether the map contains any key-value pairs. If the map is empty, it returns true; otherwise, it returns false.

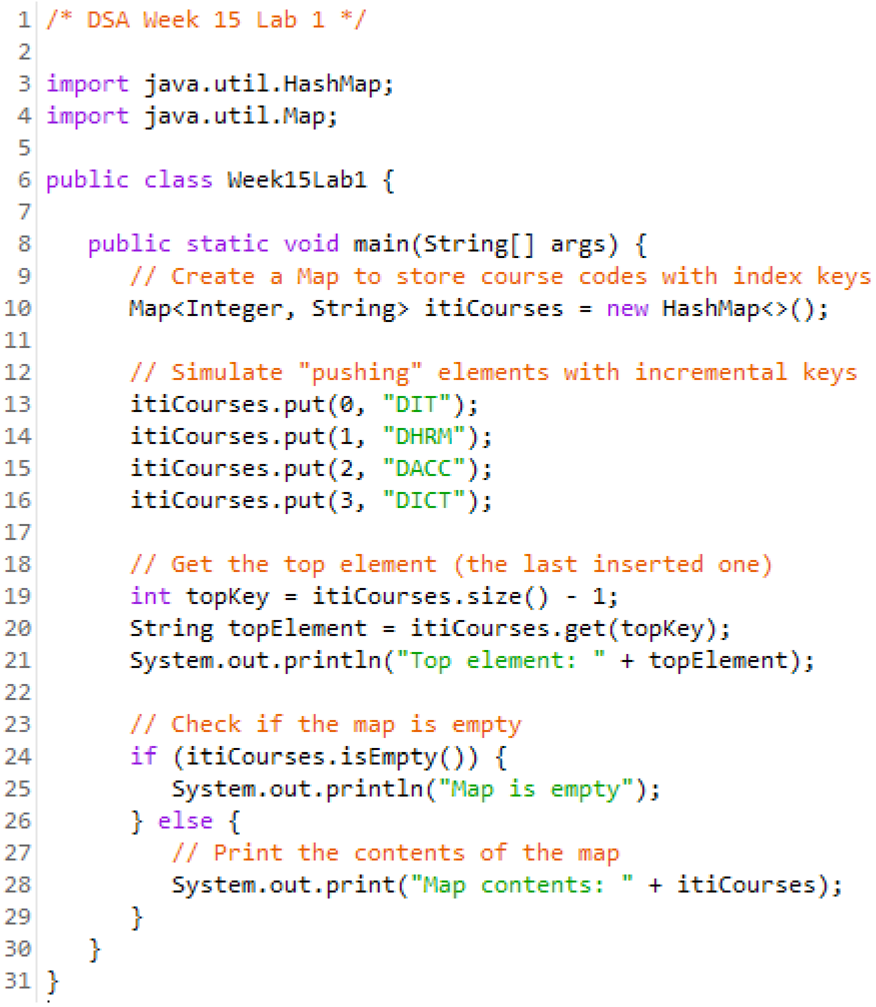
DSA Week 17 Lab Activity (Week15Lab1)

Using the lab computers create the following Java program using jGrasp!

Step 1: Login to your lab computer and create a new java file in jGrasp.

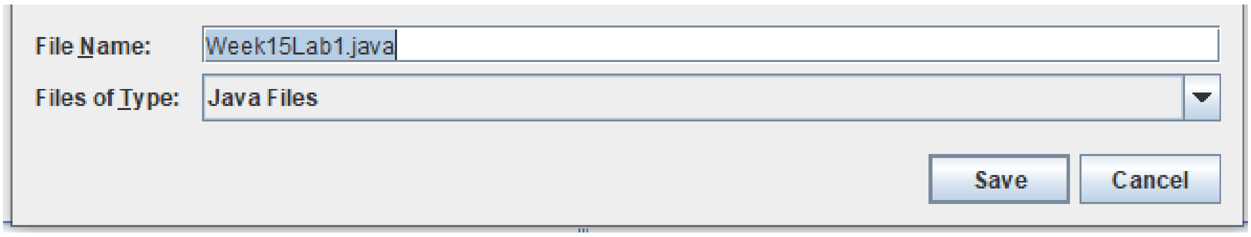


Step 2: When the window below appears. Type the following code into jGrasp.



```java
/* DSA Week 15 Lab 1 */

import java.util.HashMap;
import java.util.Map;

public class Week15Lab1 {

   public static void main(String[] args) {
      // Create a Map to store course codes with index keys
      Map<Integer, String> itiCourses = new HashMap<>();

      // Simulate "pushing" elements with incremental keys
      itiCourses.put(0, "DIT");
      itiCourses.put(1, "DHRM");
      itiCourses.put(2, "DACC");
      itiCourses.put(3, "DICT");

      // Get the top element (the last inserted one)
      int topKey = itiCourses.size() - 1;
      String topElement = itiCourses.get(topKey);
      System.out.println("Top element: " + topElement);

      // Check if the map is empty
      if (itiCourses.isEmpty()) {
         System.out.println("Map is empty");
      } else {
         // Print the contents of the map
         System.out.print("Map contents: " + itiCourses);
      }
   }
}
```

Step 3: Go to **file/save** to save your java program as **Week15Lab1**



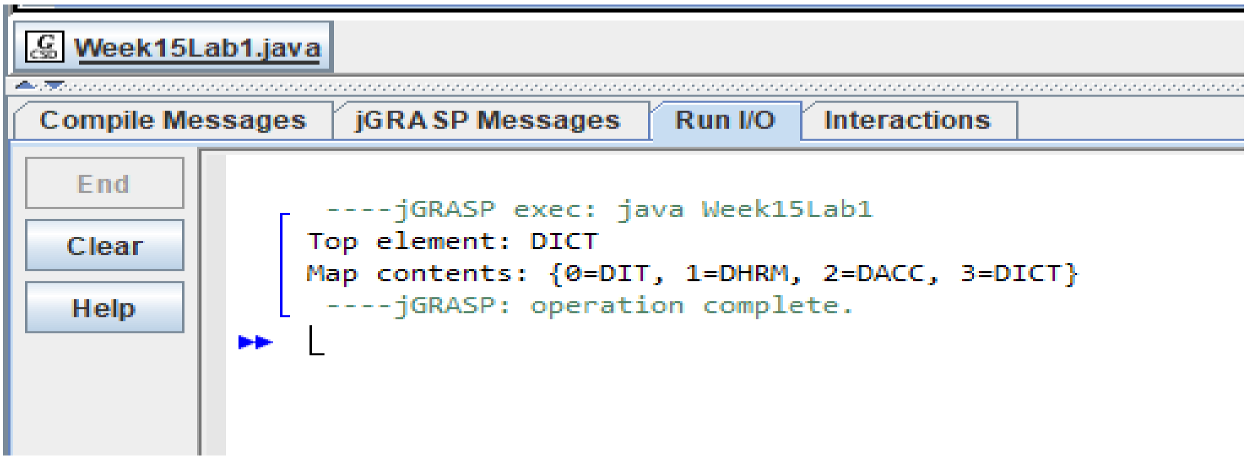
File Name: Week15Lab1.java

Files of Type: Java Files

Step 4: After saving, compile (**click on compile icon or on your keyword hold Ctrl + B**) to check for syntax errors.

Step 5: If compiling is successfully then run (**click on the find and run main method icon or on your keyboard hold Ctrl + R**) your program.

Step 6: If run is successful then you should see the following output in the console



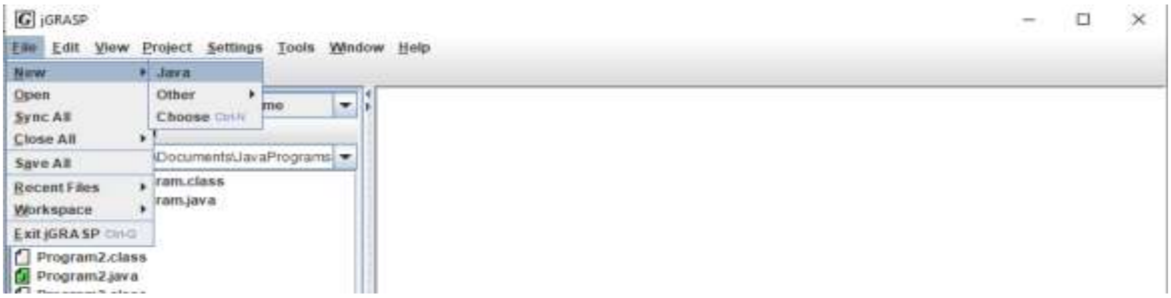
```
 ----jGRASP exec: java Week15Lab1
Top element: DICT
Map contents: {0=DIT, 1=DHRM, 2=DACC, 3=DICT}
 ----jGRASP: operation complete.
```

Step 7: Week15Lab1 Completed! Save your file for future Java lab activities.

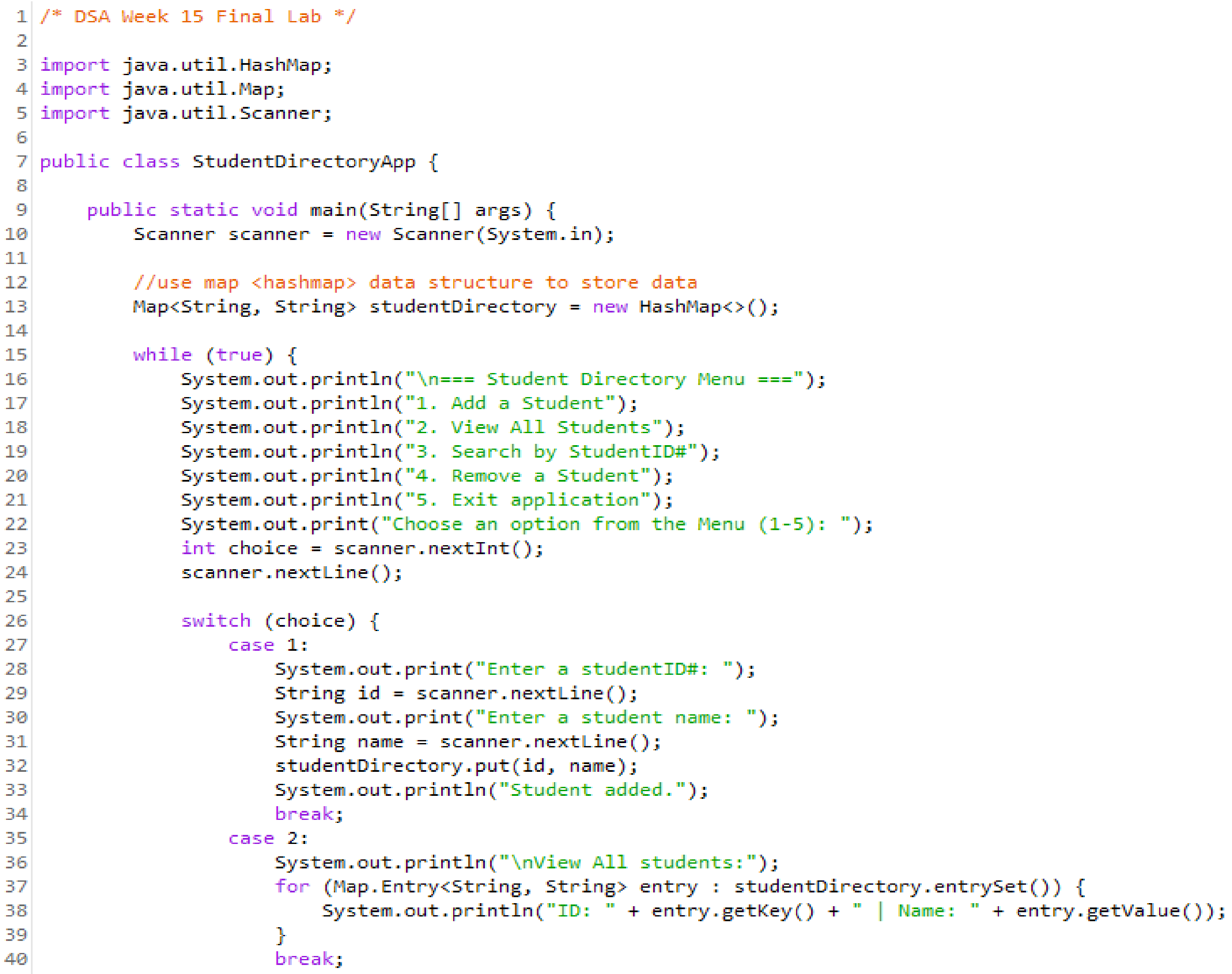
DSA Week 17 Lab Activity (StudentDirectoryApp)

Using the lab computers create the following Java program using jGrasp!

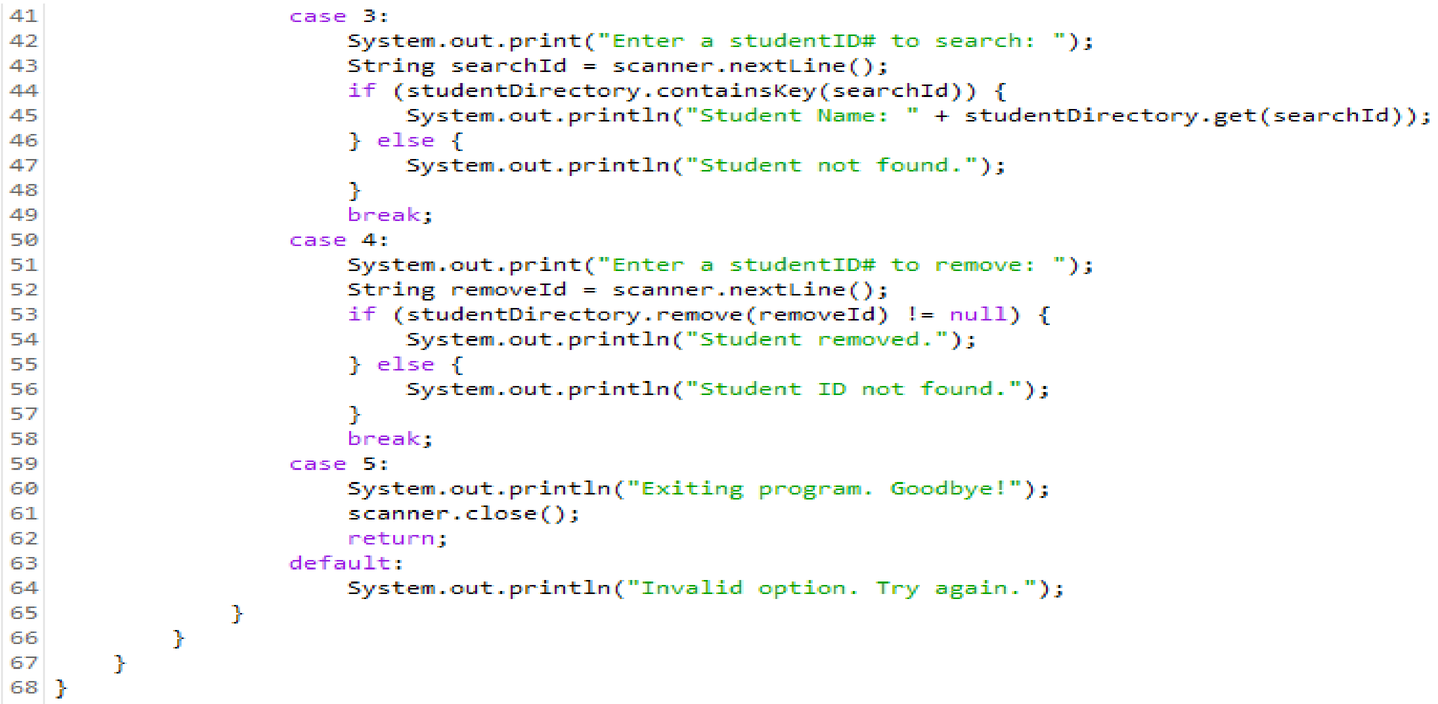
Step 1: Login to your lab computer and create a new java file in jGrasp.



Step 2: Type the following code into jGrasp.

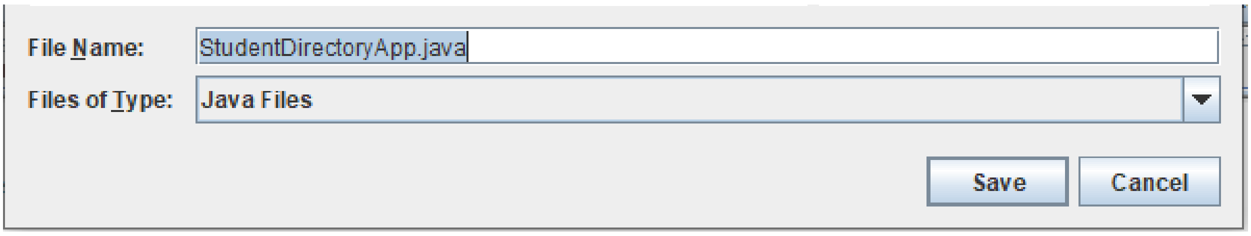


```java
/* DSA Week 15 Final Lab */

import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class StudentDirectoryApp {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        //use map <hashmap> data structure to store data
        Map<String, String> studentDirectory = new HashMap<>();

        while (true) {
            System.out.println("\n=== Student Directory Menu ===");
            System.out.println("1. Add a Student");
            System.out.println("2. View All Students");
            System.out.println("3. Search by StudentID#");
            System.out.println("4. Remove a Student");
            System.out.println("5. Exit application");
            System.out.print("Choose an option from the Menu (1-5): ");
            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    System.out.print("Enter a studentID#: ");
                    String id = scanner.nextLine();
                    System.out.print("Enter a student name: ");
                    String name = scanner.nextLine();
                    studentDirectory.put(id, name);
                    System.out.println("Student added.");
                    break;
                case 2:
                    System.out.println("\nView All students:");
                    for (Map.Entry<String, String> entry : studentDirectory.entrySet()) {
                        System.out.println("ID: " + entry.getKey() + " | Name: " + entry.getValue());
                    }
                    break;
```



```java
                case 3:
                    System.out.print("Enter a studentID# to search: ");
                    String searchId = scanner.nextLine();
                    if (studentDirectory.containsKey(searchId)) {
                        System.out.println("Student Name: " + studentDirectory.get(searchId));
                    } else {
                        System.out.println("Student not found.");
                    }
                    break;
                case 4:
                    System.out.print("Enter a studentID# to remove: ");
                    String removeId = scanner.nextLine();
                    if (studentDirectory.remove(removeId) != null) {
                        System.out.println("Student removed.");
                    } else {
                        System.out.println("Student ID not found.");
                    }
                    break;
                case 5:
                    System.out.println("Exiting program. Goodbye!");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid option. Try again.");
            }
        }
    }
}
```

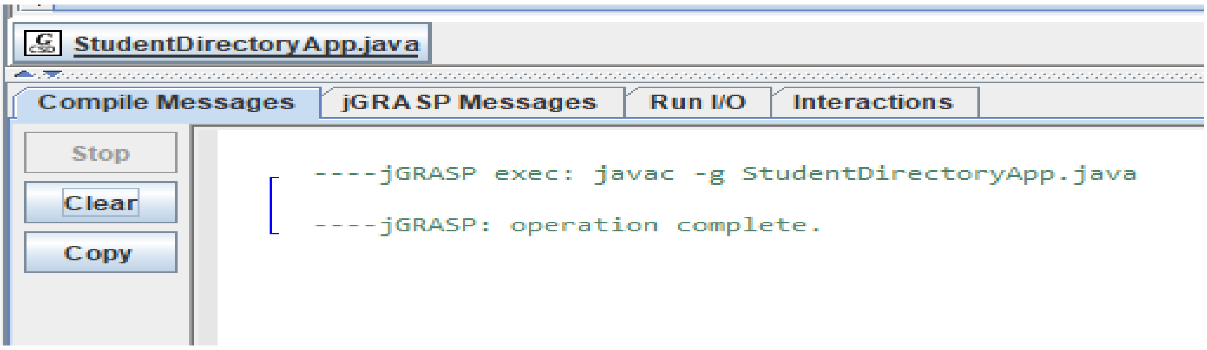
Step 3: After coding the program, go to **file/save** to save your java program as **StudentDirectoryApp**



File Name: StudentDirectoryApp.java

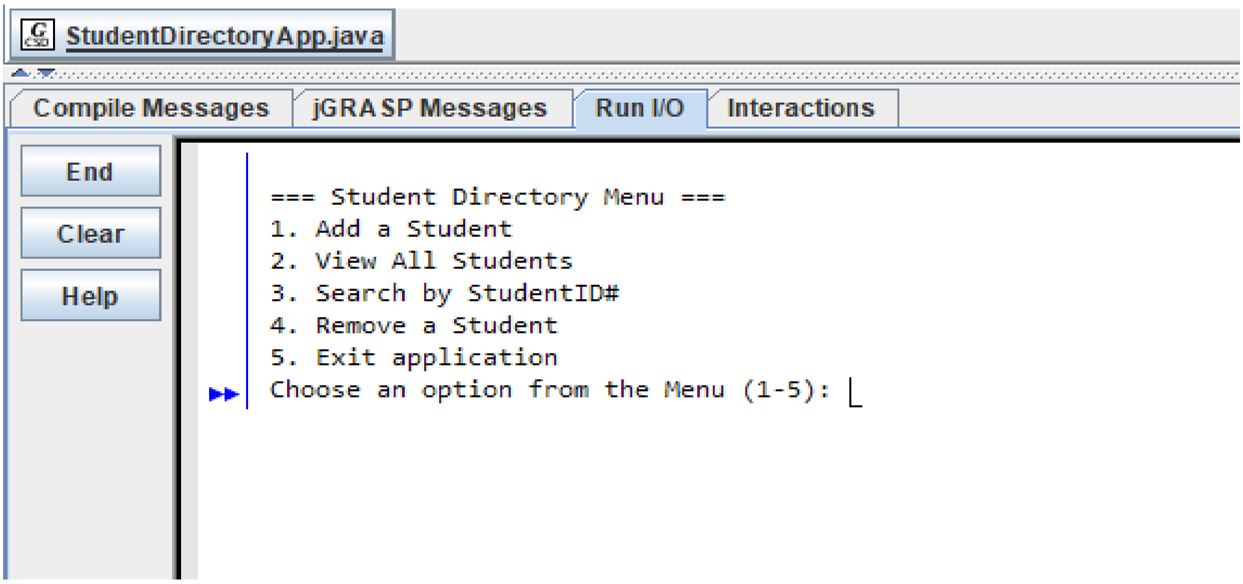
Files of Type: Java Files

Step 4: After saving, compile (**click on compile icon or on your keyword hold Ctrl + B**) to check for syntax errors.



```
----jGRASP exec: javac -g StudentDirectoryApp.java

----jGRASP: operation complete.
```

Step 5: If compiling is successfully then run (**click on the find and run main method icon or on your keyboard hold Ctrl + R**) your program.



```
=== Student Directory Menu ===
1. Add a Student
2. View All Students
3. Search by StudentID#
4. Remove a Student
5. Exit application
Choose an option from the Menu (1-5):
```

Step 6: If successful your program should display an output like shown in the screenshot above.

Step 7: Week17Lab2 Completed! From the menu type a number from 1 to 5 to perform actions as per the menu listed. This is the final lab activity for 4009 DSA.