**CIS 162 Lab8**

**ArrayLists of Objects**

**Objectives**

After completing this lab, you should be able to:

* *write* short problem solutions involving ArrayLists of objects

**Step 1: Create a New Project called Lab 8**

**Step 2: Download File**

* Download the ***students.txt*** file and save it in the same folder that contains the BlueJ project folder created for this lab, one level outside of the project files. The file has more than 2 entries!

**Step 3: Student Class**

* Create a new class called **Student.** This class has the following instance variables: a **first** name (String), **last** name (String), student **id** (int) and **gpa**(double).
* Write a constructor with parameters for each instance variable.
* Write **get** methods for each instance variable.
* **public String toString()** – Return a nicely formatted String that includes all four instance variables. (See example below)

Jenna Grand (12245) 3.8

**Step 4: Course Class**

* Create a new class called **Course.** This class has 1 instance variable: **students** as an **ArrayList** of **Student** type elements.
* Write a constructor that instantiates the ArrayList to hold Students.
* **public** **void** **readStudentData(String filename)** - open the file and read the data. (Code given later in this document. – you will need to write two statements – see the code for the TO DO comments).
* **public String chooseFile()**- creates a dialog for choosing a file. (Code given later in this document.)
* **public** **void** **printRoster()** – print all of the students on separate lines.
* **public ArrayList<Student> probationList()** – return an ArrayList of all students with a GPA below 2.0. The ArrayList will be empty if no student is on probation.
* **public Student findHighestGrade()** – return the student with the highest GPA. Assume there is at least one Student in the Course. You will just return one of the students with the highest GPA if there are two or more students with the same highest GPA.
* **public boolean searchId(int studentId)** – returns true if a student with the student id entered as parameter was found in the ArrayList otherwise returns false.

**Step 5: Create the CourseDriver Class or include the main method in the Course class.**

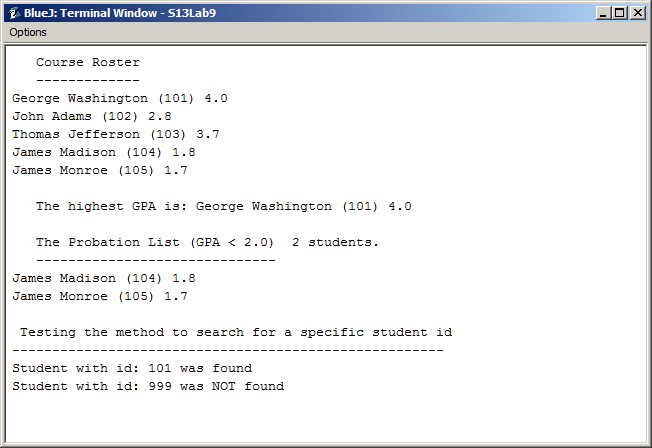
**Main method**

1. Declare and instantiate a **Course** type object (give it the name **gvsu**)
2. Select the file to read (instruction given below)

**String filename = gvsu.chooseFile();**

1. Read the data file – invoke the **readStudentData(filename)** method
2. Print the roster – invoke the **printRoster()** method
3. Find and print the highest GPA – invoke the **findHighestGrade()** method
4. Print the probation list – invoke the **probationList()** method before printing.
5. Print a message indicating if a specific student was found - Invoke the **searchId** method with parameter 101.
6. Print a message indicating if a specific student was found - Invoke the **searchId** method with parameter 999.

**Sample Results**



**Sample Data (comma delimited)**

|  |
| --- |
| George,Washington,101,4.0  John,Adams,102,2.8  Thomas,Jefferson,103,3.7  James,Madison,104,1.8  James,Monroe,105,1.7 |

**Sample Method to Read a Text File**

The following method reads from a text data file one line at a time. The data file for this example includes the last name, first name, id and a GPA on each line separated by commas.

**Sample Code**

public void readStudentData(String filename)

{

String info;

Student s;

try

{

// open the data file

Scanner fileReader = new Scanner(new File(filename));

Scanner lineReader;

// continue while there is more data to read

while(fileReader.hasNext()) {

// read one line of data

info = fileReader.nextLine();

lineReader = new Scanner(info);

lineReader.useDelimiter(",");

// read the items one at a time

String last = lineReader.next();

String first = lineReader.next();

int ID = lineReader.nextInt();

double gpa = lineReader.nextDouble();

// instantiate s as an object of the student class - TO DO

// add the student s to the ArrayList - TO DO

}

}

catch(FileNotFoundException error)

{

// could not find file

System.out.println("File not found ");

}

catch(Exception error)

{

// problem reading the file

System.out.println("Oops! Something went wrong.");

}

}

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*  \* Returns the filename, including the file path, selected by the user.  \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  public String chooseFile()  {  String filename = "";  JFileChooser fileChooser =  new JFileChooser(new File(System.getProperty("user.dir") + "\\\\.."));  fileChooser.setDialogTitle("Student files");  fileChooser.setFileSelectionMode(JFileChooser.FILES\_AND\_DIRECTORIES);  int status = fileChooser.showOpenDialog(null );  if (status == JFileChooser.APPROVE\_OPTION)  {  filename = fileChooser.getSelectedFile().getAbsolutePath();  }  return filename;  } |

Include the following import statements in your Course Class:

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.\*;

import java.util.\*;

import java.io.\*;

**Grading Criteria**

This lab is worth a possible 10 points. Show your work to the instructor or lab assistant.