For full credit you MUST create a design document and follow steps 1-4 of the following design

**Five-part program development** **process - 12.3 Application: Printing an invoice**

1.Gather requirements - Be sure you understand every detail of the requirement

2.Use CRC cards to find classes, responsibilities, and collaborators \_ Steps 1 and 2 in HOW To 12.1

3.Use UML diagrams to record class relationships - Step 3 in HOW To 12.1 and special topic 12.1, 12.2 and 12.3

4.Use javadoc to document method behavior

5.Implement your program

## Problem Statement

Your local university is going to hold an election for president of the student council. For purposes of confidentiality, the election committee wants to computerize the analysis of the voting. The committee is looking for someone to write a program to analyze the data and to report the winner. Let’s write a program to help the election committee.

The university has four major divisions. For the election, the four divisions are labeled as Department 1, Department 2, Department 3, and Department 4. The voting is reported directly to the election committee in the following form:

firstName lastName DepartmentNumber numberOfVotes

The election committee wants the output in the following tabular form:

Election Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Votes By Department | | | | | |
| Candidate Name | Dept#1 Dept#2 | Dept#3 | Dept#4 |  | Total |
| Sheila Bower | 23 70 | 133 | 267 |  | 493 |
| Danny Dillion  .  .  . | 25 71 | 156 | 97 |  | 349 |
| Winner: ???, | Votes Received: | ??? |  |  |  |

Total votes polled: ???

The names of the candidates must be in alphabetical order and dept# has to be the name of department in the output.

For this program, we assume that six candidates are seeking the student council’s president post. This program can be enhanced to handle any number of candidates.

The data is provided in four files.

1. studentInfo.txt, consists of students information in the following form(comma separated, csv file):

studentID, firstName lastName,streetAddress,City, State, zip,phoneNumber,email, major, GPA

1. candData.txt, consists of the names of the candidates seeking the president’s post. The names of the candidates in this file are in no particular order. (space separated)
2. voteData.txt, each line consists of the voting results in the following form:

firstName lastName departmentNumber numberOfVotes

That is, each line in the file voteData.txt consists of the candidate’s name, the Department number, and the number of votes received by the candidate in that Department. (space separated)

1. departments.txt consists of the department number along with department name in the following fom:

dept# departmentName officeNumber (space separated)

You must create your own data files.

Examples:

1. studentInfo.txt (comma separated, csv file):

12345, Amar, Gupta, 234 Somewhere Lane # 34, Pasadena, 91106, 626-285-3636, [jj@gmail.com](mailto:jj@gmail.com), CS, 3.85

Use Microsoft Excel compatible program to create this file then save it as CSV file

1. candData.txt (space separated)

Amar Gupta

Mike Rizvi

Lisa Fisher

The names of the candidates in this file are in no particular order.

1. voteData.txt (space separated)

Amar Gupta 2 34

Mike Rizvi 1 56

Lisa Fisher 2 56

Peter Lamba 1 78

Danny Dillion 4 29

Sheila Bower 4 78

.

.

.

The first line indicates that Amar Gupta received 34 votes from Department 2.

1. departments.txt (space separated)

1 Biology C345

2 Computer Science R312

## CLASS DESIGN

Follow directions in Chapter 12 to create the following classes . Be sure you have all of mutators and accessors methods along with equals, compareTo and toString methods

1. Person class which contains personal information
2. Student class inherited from person class with additions to student specific info such as ID
3. Candidate will be a subclass of student with addition of arrrayList of votesByDepartment and totalVotes. It will also contains static constant variable NO\_OF\_DEPARTMENTS = 4

To set the votes of a particular Department, the Department number and the number of votes are passed as parameters to the method setVotes. Because an array index starts at 0, Department 1 corresponds to the array component at position 0, and so on.

To update the votes for a particular Department, the Department number and the number of votes for that Department are passed as parameters. updateVotesByDepartment

1. Department class
2. Any additional classes you think you may need for this project

From the output, it is clear that the program must organize the voting data by Department and calculate the total votes received for each candidate and polled for the election. Furthermore, the names of the candidates must appear in alphabetical order.

The main component of this program is a candidate. Therefore, first we design the

class Candidate to implement a candidate object.

Do not use any hardcoded numbers. Use static final for any hardcoded number, such as number of candidates, in your program

Use chapter 14 for sorting options

## UML DESIGN

Include UML design for each class and overall relationships between classes – See chapter 12

## GUI DESIGN

Follow GUI designs especially chapters 11 and 20 to create GUI for this project . data entries and data displaies should be in the form of GUI. You will read all data from te files. Then you display them on your GUI for further adjustments and updates. At the end of your program you should save those data to files

Sample Run:

**Election Results**

**Votes By Department**

**Winner: Sheila Bower, Votes Received: 493**

**Total votes polled: 2216**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Candidate Name |  | Biology |  | Computer Science |  | Math |  | Psychology |  | Total |
| Sheila Bower |  | 23 |  | 70 |  | 133 |  | 267 |  | 493 |
| Danny Dillion |  | 25 |  | 71 |  | 156 |  | 97 |  | 349 |
| Lisa Fisher |  | 110 |  | 158 |  | 0 |  | 0 |  | 268 |
| Amar Gupta |  | 75 |  | 34 |  | 134 |  | 0 |  | 243 |
| Peter Lamba |  | 285 |  | 56 |  | 0 |  | 46 |  | 387 |
| Mike Rizvi |  | 112 |  | 141 |  | 156 |  | 67 |  | 476 |