**Part 1**

|  |
| --- |
| Student  **Explanation:**  firstName 🡪 Student’s first name  lastName: 🡪 Student’s last name  mathMark: 🡪 Student’s math mark  engMark: 🡪 Student’s English mark  compMark: 🡪 Student’s computer mark  stuNum: 🡪 Student number  maxMark: 🡪 Maximum possible mark  minMark: 🡪 Minimum possible mark  Student() 🡪 Creates object with all parameters  Student() 🡪 Creates object with all parameters except marks  Student() 🡪 Creates object with only student number  setMax()🡪 Sets maximum mark  setMin()🡪 Sets minimum mark  convertToLetter() 🡪 Converts percentage grade to letter  convertToGPA() 🡪 Converts letter grade to GPA  setMath() 🡪 Sets math mark  setEng() 🡪 Sets english mark  setComp() 🡪 Sets computer mark  calcAvg()🡪 Calculates average mark  calcGPA()🡪 Calculates GPA mark  toString()🡪 Returns object  The rest of get methods return corresponding values |
| -firstName: String  -lastName: String  -mathMark: double  -engMark: double  -compMark: double  -stuNum: int  +maxMark: double  +minMark: double |
| Student(fn: String, ln: String, mM: double, eM: double, cM: double, sNum: int)  Student(fn: String, ln: String, sNum: int)  Student(sNum: int)  +setMax(max: double): boolean  +setMin(min: double): boolean  +convertToLetter(mark: double): String  +convertToGPA(mark: String): double  +getFName(): String  +getLName(): String  +getMMark(): double  +getEMark(): double  +getCMark(): double  +getStuNum(): int  +getMax(): double  +getMin(): double  +setMath(math: double): boolean  +setEng(English: double): boolean  +setComp(computer: double): boolean  +calcAvg(m: double, e: double, c: double): double  +calcGPA(m: double, e: double, c: double): double  +toString(): String |

**Pseudo-code**

Read in a text file (studentdata.txt):

Open studentdata.txt file

Read data line by line

Determine the number of students (size array):

Create counter that starts at 0

Read txt file

Add 1 for every line read in file

Subtract one from counter at the end to account for a title line

Split the data:

Read line from text file

Create array to store parts of a single line from the text file

Use regex to separate by commas and spaces

Store pieces of line into array

Repeat until no more lines are available

Determine the correct Student constructor to use:

Read line from text file

Split data into array

If none of the pieces of data are null, use first constructor with all parameters

If marks from the text file line are missing, use constructor that doesn’t require marks

If all data except for the student number is missing, use constructor that only requires student number

Repeat until no more lines are available

Create an array of Student objects:

Determine number of students in text file

Create object array “Students” with size equal to number of students

**Part 2**

**Pseudo-code**

Get student number

Get object array

Check student number of first object

If student number searched for matches, return location in object array

If student number doesn’t match, retry for next object in object array

Repeat until student number location is found or if all object have been checked

Return location or -1 if location has not been found