Specs for GradeKeeper

Important notes:

1. There will be 2 members per group.
2. The grade for this MP will be 30% Milestone 1 and 70% Milestone 2
3. The Milestone 1 presentation is on 4/24 tentatively
4. The Milestone 2 presentation will be on the finals week.

Description: The GradeKeeper is a program that keeps score of students’ grades in class. The GradeKeeper keeps the following attributes of students: student number, name, grades.

Create a command line interpreter with the following specs:

1. When starting the program, the user can supply a database file as an argument. If no argument is supplied, the program would open/create a file “gradeKeeper.txt”

Ex.

```
gk “/files/grades.txt“
gk>
```

2. If the file was successfully opened, the program would start. Otherwise, the program would end and would show the error to the user.

3. Commands to implement:

list [keyword]: If keyword is not supplied, the command will display all the students in the file. If keyword is supplied, the command will display students when their names or student numbers match the keyword (matching is not case sensitive). The average grade (2 decimal places) of the student should also be displayed.
Note: keywords containing one or more words should be enclosed in double quotes.

Ex.

```
gk> list

Student Number: 2017-99999
Name: Berwin T. Yu
Exam Grades: 100, 90, 80
Average: 90.00

Student Number: 2017-11111
Name: Berwin A. Yu
Exam Grades: 98, 100, 56, 70
Average: 81.00
Student Number: 2017-12345

Name: John
Exam Grades: 100, 99.5, 90.25
Average:96.58

gk> list Ber

Student Number: 2017-99999
Name: Berwin T. Yu
Exam Grades: 100, 90, 80
Average: 90.00

Student Number: 2017-11111
Name: Berwin A. Yu
Exam Grades: 98, 100, 56, 70
Average: 81.00

gk> list -1

Student Number: 2017-11111
Name: Berwin A. Yu
Exam Grades: 98, 100, 56, 70
Average: 81.00

Student Number: 2017-12345
Name: John
Exam Grades: 100, 99.5, 90.25
Average:96.58
```

```
gk> list Berwin A.

Student Number: 2017-99999
Name: Berwin T. Yu
Exam Grades: 100, 90, 80
Average: 90.00

Student Number: 2017-11111
Name: Berwin A. Yu
Exam Grades: 98, 100, 56, 70
Average: 81.00

gk> list “Berwin A.”

Student Number: 2017-11111
Name: Berwin A. Yu
Exam Grades: 98, 100, 56, 70
Average: 81.00
```

add [student number] [name] [grades]: Adds a student to the database with the parameters supplied. Student number should be of the format xxxx-xxxxx where x is a 1-digit number. Student number should also be unique: cannot use student numbers that exist. Name should be enclosed in double quotes if it has spaces. Grades are optional parameters and they should be separated by commas without spaces. Grades range from 0-100. Decimals are accepted for grades. All input grades will be rounded off to 2 decimal places (if applicable)

```
gk> add 2017-12345

Error! Not enough parameters

gk> add 2017-12345 Berwin

Error! 2017-12345 is already taken.
gk> add 2017-22222 Berwin

gk> add 2017-333 John

Error! Invalid student number.

gk> add 2017-33333 John 100,90,100
gk> add 2017-44444 Matthew 100,hj,88

Error! Invalid grade.
```

help [command]: If command is not supplied, the command displays all the valid commands. If command is supplied, print out detailed information about a command alongside its usage.

exit: Exits the program

------------------------------Milestone 1-----------------------------

delete [student number]: Deletes the record of the student that has the student number specified. Student number is a required field.

grade [student number] [grades]: Add grades to the specific student number. Grades should be separated by commas without spaces. Grades range from 0-100. Student number is optional. If student number is supplied, then grades should also be supplied. If no parameters are supplied, the program would display each student number and name and let the user choose which record will be added a grade. After that the user can supply the grades. The user can input return at any point if he/she want to go back.

Ex.

```
gk> grade 2017-11111 100,78.5

gk> grade 2017-11115 100,78.5

Error! Student number does not exist.

gk> grade 2017-11115 100, 5

Error! Wrong format for grades.

gk> grade 2017-11111

Error! Incomplete parameters.
```

```
gk> grade

[1] 2017-99999 Berwin T. Yu
[2] 2017-11111 Berwin A. Yu
[3] 2017-12345 John
[4] 2017-22222 Berwin
[5] 2017-33333 John

Input the record to want to add grade: a
Error! Record does not exist.
Input the record to want to add grade: 6
Error! Record does not exist.
Input the record to want to add grade: 5

Input grades: 100,101
Error! Record does not exist.
Input grades: 90.999,95

gk> grade

[1] 2017-99999 Berwin T. Yu
[2] 2017-11111 Berwin A. Yu
[3] 2017-12345 John
[4] 2017-22222 Berwin
[5] 2017-33333 John

Input the record to want to add grade: return

gk>
```

edit: Edits a grade of a student. The program would display each student number and name and let the user choose which record will be edited. After that the user chooses which grade will be edited. Finally, the user will input the grade. At any point in the process above, the user can input return to go back to the command line.

Ex.

```
gk> edit

[1] 2017-99999 Berwin T. Yu
[2] 2017-11111 Berwin A. Yu
[3] 2017-12345 John
[4] 2017-22222 Berwin
[5] 2017-33333 John

Input the record to want to edit: 1

[1] 100
[2] 90
[3] 80

Input the exam number you want to edit: j
Error! Record number does not exist.
Input the exam number you want to edit: 3
Change exam 3 to: return

gk>
```

stat: Displays the mean, median, and mode, standard deviation and range of all the grades of all students.

Additional requirements to be followed:
1. Entering a blank or white spaces should not throw an error.
2. Commands should be trimmed.

```
   Ex. gk>       list     Ber
```

The command above is considered a valid input
3. Update the help command to include the new commands