name\_grades.txt looks like:

\*do not put actual words just the grades for that category in this specific order\*

Labs// 20 20 20 20 20 20 20 20 20 20 = 200

Assignments// 50 50 50 50 = 200

Projects// 150 350 = 500

Exam// 100 = 100

TOTAL: 200+200+500+100=1000

\*\*Main.CPP\*\*

//take 1 CLA

* Input file (name\_grades.txt)

//ask user which category they want to see

* Read the name to the line of grade
* Be specific “individual”
  + name\_of\_category
  + Num
* “category” ex: lab Assignments
* “Course” all the data of the file totals

//based on read in information

* Create instance of class Grade\_Book
* //read file

-while data don’t end with “ “ read the next line

-separate grades into vectors

- Call function

// print output

\*\*Grade\_Book.h\*\*

Private:

std::vector<int> \*labs;

std::vector<int> \*assignments;

std::vector<int> \*projects;

std::vector<int> \*Exam;

Int total;

public :

Grade\_Book();

Void read\_file(std::string filename); //amber

Int individual();//Gio

Int category(); //Jun

Return all output and

Int course(); // mike?

:

\*\*Grade\_Book.cpp\*\*

//create math function to calculate total grade

// Loop function to return all grades and sum of the list

**EXAMPLE EXECUTION CASES:**

**// mike**

./grades [input\_file\_name] [name\_category] [task\_number] // individual case

If [name\_category] == ‘individual’:

If [name\_category] == ‘category’:

If [name\_category] == ‘course’:

If [name\_category] == BAD\_INPUT:

Catch bad input and tell user to run ./grades help

***// recommends user to run ./grades help***

./grades [bad\_command\_line\_input]

***// displays proper command usage (error handling)***

./grades help

Return be like

Input “lab1”

Output “10 10000”