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”