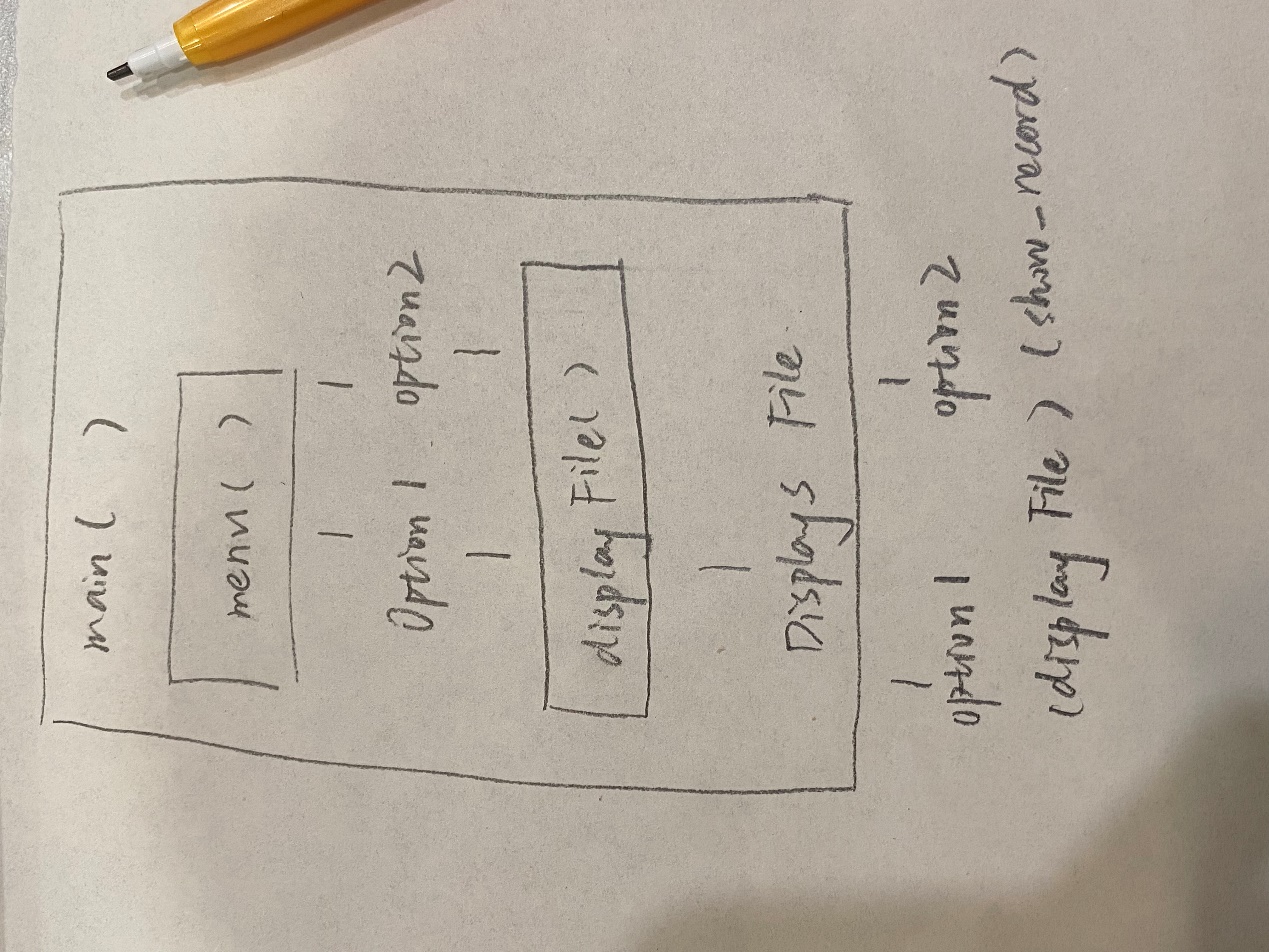
a. output: SyntaxError: multiple statements found while compiling a single statement



3.

infile = open (infileName, 'r') :

infile: This is a variable name used to store file objects. After this statement is executed, infile will reference the opened file.

open: This is a built-in function for opening files. In this case, it opens a file with the 'r' parameter (which means read-only).

infileName: This is the variable or constant that contains the file name, indicating the name of the file to be opened.

'r' : This is the file open mode, which means that the file is opened in read-only mode. This means that the program can read the contents of the file but cannot write to it.

4.positive:

Reusability: The breakdown of functionality into smaller functions allows these functions to be reused in different contexts. If a function needs to be used in multiple parts of the program, the same function can be called in multiple places, improving code reuse.

Debugging and testing: small functions are easy to debug and test. When there is an error in the program, it is easier to trace the problem to a specific function. This allows you to quickly locate and fix errors without having to look at the entire program.

Team collaboration: In large projects, different team members can be responsible for different functions without interfering with each other. This modular design makes team collaboration more efficient, and everyone can focus on the part they are responsible for.

5.

#display the student name given student id

def display\_student\_name(file\_path, student\_id):

with open("C:\Users\Ciel\Desktop\class.txt", 'r') as file:

lines = file.readlines()

for line in lines:

parts = line.split(',')

if int(parts[0]) == int(student\_id):

print(parts[1])

break

#display the student mark given student id

def display\_student\_mark(file\_path, student\_id):

with open("C:\Users\Ciel\Desktop\class.txt", 'r') as file:

line = file.readlines()

for line in lines:

part = line.spilt(',')

if int(parts[0]) == int(student\_id):

print(parts[2])

break

#display the maximum mark

def display\_max\_mark(file\_path):

with open("C:\Users\Ciel\Desktop\class.txt", 'r') as file:

lines = file.readlines()

max\_mark = float('-inf')

for line in lines:

parts = line.split(',')

mark = float(parts[2])

if mark > max\_mark:

max\_mark = mark

print("Maximum mark: ", max\_mark)

break

#display the minimum mark with student id

def display\_min\_mark\_and\_student\_id(file\_path):

with open("C:\Users\Ciel\Desktop\class.txt", 'r') as file:

lines = file.readlines()

min\_mark = float('inf')

min\_student\_id = 0

for line in lines:

parts = line.split(',')

mark = float(parts[2])

if mark < min\_mark:

min\_mark = mark

min\_student\_id = int(parts[0])

print("Minimum mark: ", min\_mark)

print("Student ID: ", min\_student\_id)

break

# function to display all the students who has a pass mark

def pass\_mark(inf, studentID):

for line in inf:

mark = extract\_student\_mark(line)

if mark >= pass\_threshold: # Define pass threshold as needed

print ("Student ID:", extract\_student\_ID(line), "passed with a mark of", mark)

return

# function to Display student list in grade groups

def show\_group(inf, studentID):

function show\_group(inf):

grade\_groups = {"HD": [], "D": [], "C": [], "P": [], "N": []}

for line in inf:

mark = extract\_student\_mark(line)

student\_ID = extract\_student\_ID(line)

if mark >= 80:

grade\_groups["HD"].append(student\_ID)

elif mark >= 70:

grade\_groups["D"].append(student\_ID)

elif mark >= 60:

grade\_groups["C"].append(student\_ID)

elif mark >= 50:

grade\_groups["P"].append(student\_ID)

else:

grade\_groups["N"].append(student\_ID)

for group, students in grade\_groups.items():

print ("Grade Group", group, ":", students)

return

def stub():

print("This function is not yet available")

return

import string

# Student data

student\_data = [

student\_data.append(“35032101\tPhil\t92”)

student\_data.append(“910291\tSue\t45”)

student\_data.append(“32891820\tJim\t21”)

student\_data.append(“23910283\tSally\t98”)

student\_data.append(“47293827\tSam\t23”)

student\_data.append(“34139281\tJoan\t55”)

student\_data.append(“87392010\tJohn\t69”)

student\_data.append(“56373847\tPhil\t45”)

student\_data.append(“62527282\tLisa\t87”)

student\_data.append(“38274738\tJack\t81”)

student\_data.append(“45748380\tAnn\t56”)

student\_data.append(“19284729\tPat\t71”)

student\_data.append(“82938402\tTara\t50”)

student\_data.append(“67492829\tTadhg\t83”)

student\_data.append(“12083849\tRoger\t56”)

student\_data.append(“57302843\tTom\t28”)

student\_data.append(“34837900\tRose\t45”)

student\_data.append(“98278388\tTim\t62”)

student\_data.append(“48204803\tKaye\t78”)

student\_data.append(“34924920\tLisa\t84”)

]