Final program:

User Manual; To use the program(s), it will ask you for an input and then the program is designed so that it will take the data from the provided example of client sheets to run the program and give the output in another file.

For year 7-10's Students:

import csv# Import the end of term year 9's marks import math import openpyx1

def get\_file\_input(): file\_path = input("Please enter your excel file: ")

try:  
 workbook = openpyx1.load\_workbook(file\_path)  
 sheet = workbook.activate  
   
 sheet.cell(row=row, column =10, value=calculate\_gpa())  
  
   
 workbook.save("Updated\_" + file\_path)

class Student:

def \_\_**init\_\_**(self,studentcode,studentname,englishmarks,hassmarks,mathmarks,sciencemarks,englishmarks2,hassmarks2,mathsmarks2,sciencemarks2):

self.studentcode = studentcode

self.studentname = studentname

self.englishmarks = int(englishmarks)

self.hassmarks = int(hassmarks)

self.mathmarks = int(mathmarks)

self.sciencemarks = int(sciencemarks)

self.englishmarks2 = int(englishmarks2)

self.hassmarks2 = int(hassmarks2)

self.mathsmarks2 = int(mathsmarks2)

self.sciencemarks2 = int(sciencemarks2)

def calculate\_gpa(self):  
 """Calculate the gpa of every student."""  
 gpa = (self.englishmarks + self.hassmarks + self.mathmarks + self.sciencemarks + self.englishmarks2 + self.hassmarks2 + self.mathsmarks2 + self.sciencemarks2)/4  
 return round(gpa)

students = []

filename = with open(mode = 'r') as file: csv\_reader = csv.reader(file)

index = 0

for row in csv\_reader:  
 index = index + 1  
 print(row)  
 if index ==1 or index == 2 or index == 3 or index == 4:  
 continue  
 for i in range(2,10):  
 if row[i] == "":  
 row[i] = 0  
   
   
 student = Student(row[0],row[1],row[2],row[3],row[4],row[5],row[6],row[7],row[8],row[9])  
 students.append(student)

storing\_gpa = []

print("number of students", len(students)) for student in students: all\_names.append(student.studentname) current\_gpa = student.calculate\_gpa() storing\_gpa.append(current\_gpa)

For year 11-12 General Students:

import csv# Import the end of term year 9's marks import openpyx1

def get\_file\_input(): file\_path = input("Please enter your excel file: ")

try:  
 workbook = openpyx1.load\_workbook(file\_path)  
 sheet = workbook.activate  
   
 sheet.cell(row=row, column =26, value=calculate\_gpa())  
 sheet.cell(row=row, column =27, value=count\_cs())  
   
 workbook.save("Updated\_" + file\_path)

class Student: def **init**(self,studentcode,studentname,eng,tec,csi,wsi,acf,vdt,tec2,app,app2,bme,bsp,cae,che,eng2,hea,vho,hpe,hum,mat,oed,pes,var):

self.studentcode = int(studentcode)

self.studentname = studentname

self.eng = int(eng)

self.tec = int(tec)

self.csi = int(csi)

self.wsi = int(wsi)

self.acf = int(acf)

self.vdt = int(vdt)

self.tec2 = int(tec2)

self.app = int(app)

self.app2 = int(app2)

self.bme = int(bme)

self.bsp = int(bsp)

self.cae = int(cae)

self.che = int(che)

self.eng2 = int(eng2)

self.hea = int(hea)

self.vho = int(vho)

self.hpe = int(hpe)

self.hum = int(hum)

self.mat = int(mat)

self.oed = int(oed)

self.pes = int(pes)

self.var = int(var)

def calculate\_gpa(self):  
 """Calculate the gpa of every student."""  
 gpa = (self.eng + self.tec + self.csi + self.wsi + self.acf + self.vdt + self.tec2 + self.app + self.app2 + self.bme + self.bsp + self.cae + self.che + self.eng2 + self.hea + self.vho + self.hpe + self.hum + self.mat + self.oed + self.pes + self.var)/4  
 return round(gpa)

students = []

filename = with open(mode = 'r') as file: csv\_reader = csv.reader(file) index = 0

for row in csv\_reader:  
 index = index + 1  
 print(row)  
 if index ==1 or index == 2 or index == 3:  
 continue  
 for i in range(2,25):  
 if row[i] == "":  
 row[i] = 0  
   
   
 student = Student(row[0],row[1],row[2],row[3],row[4],row[5],row[6],row[7],row[8],row[9],row[10],row[11],row[12],row[13],row[14],row[15],row[16],row[17],row[18],row[19],row[20],row[21],row[22],row[23],row[24])  
 students.append(student)

storing\_gpa = [] all\_subject = [] # Constants C\_threshold = 50 # Minimum average score to achieve a "C"

for student in students: storing\_gpa = calculate\_gpa() all\_subjects.append(student.eng,student.tec,student.csi,student.wsi,student.acf,student.vdt,student.tec2,student.app,student.app2,student.bme,student.bsp,student.cae,student.che,student.eng2,student.hea,student.vho,student.hpe,student.hum,student.mat,student.oed,student.pes,student.var)

def count\_cs():  
 """Checks if the student passes (based on C\_threshold) and assigns them two C's if they passed that subject"""  
 if any(all\_subject >= C\_threshold):  
 return "Two C's"  
 else:  
 return 'No C'

For year 11-12 Atar Students:

import csv import openpyx1

def get\_file\_input(): file\_path = input("Please enter your excel file: ")

try:  
 workbook = openpyx1.load\_workbook(file\_path)  
 sheet = workbook.activate  
   
 sheet.cell(row=row, column =28, value=calculate\_atar())  
 sheet.cell(row=row, column =29, value=calculate\_tea())  
 sheet.cell(row=row, column =30, value=count\_cs())  
   
 workbook.save("Updated\_" + file\_path)

for row in csv\_reader:

index = index + 1

print(row) if index ==1 or index == 2 or index == 3 or index == 4: continue for i in range(2,27): if row[i] == "": row[i] = 0

# **Open the output file for writing**

with open(output\_file, 'w', newline='') as outfile:

writer = csv.writer(outfile)

writer.writerows(rows\_without\_header) # Write the remaining rows to the output file

index = 0

for row in csv\_reader: index += 1 print(row)

import csv# Import the end of term year 12’s subject marks

class Student:

def \_\_init\_\_(self,studentcode,studentname,acc,acf,app,ara,bme,bme2,cae,che,che2,com,eng,eng2,hea,hea2,hum,hum2,mat,mat2,mat3,mat4,oed,pes,phy,psy,var):  
  
 self.studentcode = studentcode   
 self.studentname = studentname   
 self.acc = int(acc)   
 self.acf = int(acf)   
 self.app = int(app)   
 self.ara = int(ara)   
 self.ara = int(ara)   
 self.bme = int(bme)   
 self.bme2 = int(bme2)   
 self.cae = int(cae)   
 self.che = int(che)   
 self.che2 = int(che2)   
 self.com = int(com)   
 self.eng = int(eng)  
 self.eng2 = int(eng2)  
 self.hea = int(hea)   
 self.hea2 = int(hea2)   
 self.hum = int(hum)   
 self.hum2 = int(hum2)   
 self.mat = int(mat)   
 self.mat2 = int(mat2)   
 self.mat3 = int(mat3)   
 self.mat4 = int(mat4)   
 self.oed = int(oed)   
 self.pes = int(pes)   
 self.phy = int(phy)   
 self.psy = int(psy)   
 self.var = int(var)

student = []

student = Student(row[0],row[1],row[2],row[3],row[4],row[5],row[6],row[7],row[8],row[9],row[10], row[11],row[12],row[13],row[14],row[15],row[16],row[17],row[18],row[19],row[20],row[21],row[22],row[23],row[24],row[25],row[26],row[27])

students.append(student)

top\_4\_marks = []

all\_subject = []

for student in students:

top\_4\_marks = get\_top\_4\_marks() all\_subjects.append(student.acc,student.acf,student.app,student.ara,student.bme,student.bme2,student.cae,student.che,student.che2,student.com,student.eng,student.eng2,student.hea,student.hea2,student.hum,student.hum2,student.mat,student.mat2,student.mat3,student.mat4,student.oed,student.pes,student.phy,student.psy,student.var)

# **#Constants**

C\_threshold = 50 # Minimum average score to achieve a "C"

def get\_top\_4\_marks(self,):  
 """Retrieve the top 4 subject marks for a student."""  
 return sorted (self.acc,self.acf,self.app,self.ara,self.bme,self.bme2,self.cae,self.che,self.che2,self.com,self.eng,self.eng2,self.hea,self.hea2,self.hum,self.hum2,self.mat,self.mat2,self.mat3,self.mat4,self.oed,self.pes,self.phy,self.psy,self.var, reverse=True)[:4]

def calculate\_tea(): """Calculate their tea score""" return (top\_4\_marks \* 100)

# **def calculate\_atar():**

# **"""Calulate Atar using their top 4 subject marks"""**

def count\_cs(): """Checks if the student passes (based on C\_threshold) and assigns them two C's if they passed that subject"""

if any(all\_subject >= C\_threshold):

return "Two C's"

else:

return 'No C'

Testing, documentation and security:

To test these files they were modified mannier times and by the use of AI I had an idea on what the program was missing and with the teacher’s help I transformed my code into final program based on my understanding of the clients need’s. The program will run and might have errors in some places or some code missing due my unknowlegde of the completing the code but also time running out for asking for help from the teacher.

The program was a lot complex to get around and figure out the right way to do the program so I did run out of time but also not getting client feedback until the very end of the due date but she didn’t came before when she was actually going to come. It took me a week just to get the year 7-10 code because I had other assessments too but to finish the other two main harder designation of the program, I had to just hurry the entire process because of the time then went by like nothing but other personal issues too but I think I should had tried to make everything clear to the teacher and should have dedicated more time outside of class to get things done.

There were not any ‘security’ concerns but I lost the program or forgot to save that version where it had the additional comments explaining sections but the school z drive and working from home was not easy the best to work consistently.

Presentation and reflection:

The program I was going to put comments in explaining each section of the code but the sr Anita just wanted the outcome to matter. So the program uses object oriented programming where if has functions splitting the code and making it less hard to design the code but in the reflection I want to say I lacked and let the time pass by without communicating effectively with the teacher and with me getting a bad headache and the ankle sprain and Wi-Fi issues on the day before the submission I mean everything was against what I needed to have happened, but no excuses I should had dedicated a lot of time during the first week and a half.