K.R. MANGALAM UNIVERSITY

THE COMPLETE WORLD OF EDUCATION

PROGRAMMING FOR PROBLEM SOLVING USING PYTHON

B.TECH CSE CORE SECTION (A) SEMESTER - 1

COURSE CODE : ETCCPP102

ASSIGNMENT NO. : 2

ASSIGNMENT TITLE : GRADE BOOK ANALYZER

SUBMITTED BY : MANPREET KAUR

SUBMITTED TO : FEROZ AHMAD SIR

DATE OF SUBMISSION : 5 DECEMBER 2025

Name	Date modified	Type	Size
grade-book	04-12-2025 09:33	Microsoft Excel Co...	1 KB
Grade-Book-Analyzer	04-12-2025 09:41	Python Source File	7 KB
grade-book-csv	04-12-2025 09:31	Python Source File	1 KB

GRADE-BOOK-ANALYZER.PY

C:\Users\MANPREET KAUR\Desktop\PYTHON\Grade-Book-Analyzer\Grade-Book-Analyzer.py

grade-book-csv.py × Grade-Book-Analyzer.py ×

```python
# Grade-Book-Analyzer.py

# Author : Manpreet Kaur

# Date : 3-December-2025

# project : Grade-Book-Analyzer --- Analyze and Report Student Grades using Python

import csv   # importing the csv module

# Creating the welcome_message function that will print the welcome message when you run this program

def welcome_message():
    print("Welcome to Grade-Book-Analyzer !")
    print("Choose data input method : ")
    print("1. Manual input")
    print("2. Load from a CSV file")

# Creating the manual_input function which will take manual entries from the user
# and return a dictionery named students

def manual_input():
    students={}
    n=int(input("Enter the number of students : "))
    for i in range(n):
        name=input("Enter the name of the student : ")
        marks=int(input(f"Enter the marks of {name} : "))
        students[name]=marks
    return students

# Creating the load_CSV function which will import the data from a pre-existing csv file
# and return a dictionery named students

def load_CSV(filename):
    students={}
    with open(filename,newline="") as csvfile:
        reader=csv.reader(csvfile)
        for row in reader:
            name=row[0]
```

C:\Users\MANPREET KAUR\Desktop\PYTHON\Grade-Book-Analyzer\Grade-Book-Analyzer.py

grade-book-csv.py × Grade-Book-Analyzer.py ×

```python
            name=row[0]
            marks=int(row[1])
            students[name]=marks
    return students

# Creating the function calculate_avergae that will calculate the average marks of the students
# and return the average

def calculate_average(marks_dict):
    total=sum(marks_dict.values())
    count=len(marks_dict)
    average=total/count
    return round(average,2)

# Creating the function calculate_median that will calculate the median marks of the students
# and return the median

def calculate_median(marks_dict):
    marks_list=sorted(marks_dict.values())
    n=len(marks_list)
    if n%2==0:
        median=(marks_list[n//2-1]+marks_list[n//2])/2
    else:
        median=marks_list[n//2]
    return round(median,2) if isinstance(median, float) else median

# Creating the function find_max_score that will find the maximum score ,
# scored by the student and return the maximum score and the student name with max score

def find_max_score(marks_dict):
    max_score=max(marks_dict.values())
    max_students=[name for name, score in marks_dict.items() if score == max_score ]
    return max_score, max_students

# Creating the function find_min_score that will find the minimum score ,
# scored by the student and return the minimum score and the student name with min score
```

C:\Users\MANPREET KAUR\Desktop\PYTHON\Grade-Book-Analyzer\Grade-Book-Analyzer.py

grade-book-csv.py × Grade-Book-Analyzer.py ×

```python
def find_min_score(marks_dict):
    min_score=min(marks_dict.values())
    min_students=[name for name, score in marks_dict.items() if score == min_score ]
    return min_score, min_students

# Grade Assignment

# Creating a function assign_grades that will assign the grades to the students according to their score
# and add the score to a dictionery

def assign_grades(marks_dict):
    grades={}
    for student, marks in marks_dict.items():
        if marks >= 90:
            grade="A"
        elif marks >= 80:
            grade="B"
        elif marks >= 70:
            grade="C"
        elif marks >= 60:
            grade="D"
        else:
            grade="F"
        grades[student]=grade
    return grades

# Creating a function grade_distribution that will count the number of students getting a
# corresponding grade and return the grade destribution

def grade_distribution(grades_dict):
    distribution={"A":0,"B":0,"C":0,"D":0,"F":0}
    for grade in grades_dict.values():
        distribution[grade]+=1
    return distribution

# Pass / Fail Filter using list comprehension

# Creating a function filter_pass_fail that will filter all the failed and passed students
# separately and return the number of failed and passed students along with their names
```

```python
def filter_pass_fail(marks_dict):
    passed=[name for name, marks in marks_dict.items() if marks >=40 ]
    failed=[name for name, marks in marks_dict.items() if marks < 40 ]
    return passed, failed

# Results Table Printing function : that will print the final results table

def print_results_table(marks_dict, grades_dict):
    print("Name            Marks    Grade")
    print("-------------------------------------")
    for name in marks_dict:
        print(f"{name:<15}    {marks_dict[name]:<7}    {grades_dict[name]}")
    print("-------------------------------------")

# this main function will print the welcome message and take the choice from the user that
# user wants to import the csv file or write the data manually and many more things .

def main():
    welcome_message()
    choice=input("Enter your choice (1 or 2) : ")
    if choice=="1":
        students_marks=manual_input()
    elif choice=="2":
        filename=input("Enter CSV filename : ")
        students_marks=load_CSV(filename)
    else:
        print("Invalid Choice ! , Exit caused due to Invalid Choice .")

# Statistical summary : calling the functions and storing them in the variables

    avg=calculate_average(students_marks)
    med=calculate_median(students_marks)
    max_score,max_students=find_max_score(students_marks)
    min_score,min_students=find_min_score(students_marks)
```

grade-book-csv.py × Grade-Book-Analyzer.py ×

```
151  # printing the final results
152
153      print("Statistics Summary : ")
154      print(f"Average marks : {avg}")
155      print(f"Median marks : {med}")
156      print(f"Max Score : {max_score} by {','.join(max_students)}")
157      print(f"Mix Score : {min_score} by {','.join(min_students)}")
158
159  # Assigned grades : printing the grade distribution
160
161      grades=assign_grades(students_marks)
162      dist=grade_distribution(grades)
163      print("Grade Distribution : ")
164      for grade, count in dist.items():
165          print(f"{grade} : {count} students")
166
167  # Pass / Fail filtering : printing the filtered pass and failed students
168
169      passed_students, failed_students = filter_pass_fail(students_marks)
170      print(f"Passed Students ({len(passed_students)}) : {','.join(passed_students)}")
171      print(f"Failed students ({len(failed_students)}) : {','.join(failed_students)}")
172
173  # Printing the final table
174
175      print_results_table(students_marks,grades)
176
177  # User Loop to repeat or exit : that will ask user that if he want to analyse another grade book or
178  # want to exit the Grade-Book-Analuzer
179
180      while True:
181          repeat=input("Do you want to analyze another set? (yes/no) : ").strip().lower()
182          if repeat=="yes":
183              main()
184              break
185          elif repeat=="no":
186              print("Thank you for using Grade-Book-Analyzer ! ")
187              break
188          else:
189              print("Invalid input, please enter yes or no.")
190
```

```
190
191  # these lines starts the main grade analyzer program by calling the main() funtion
192
193  if __name__=="__main__":
194      main()
```

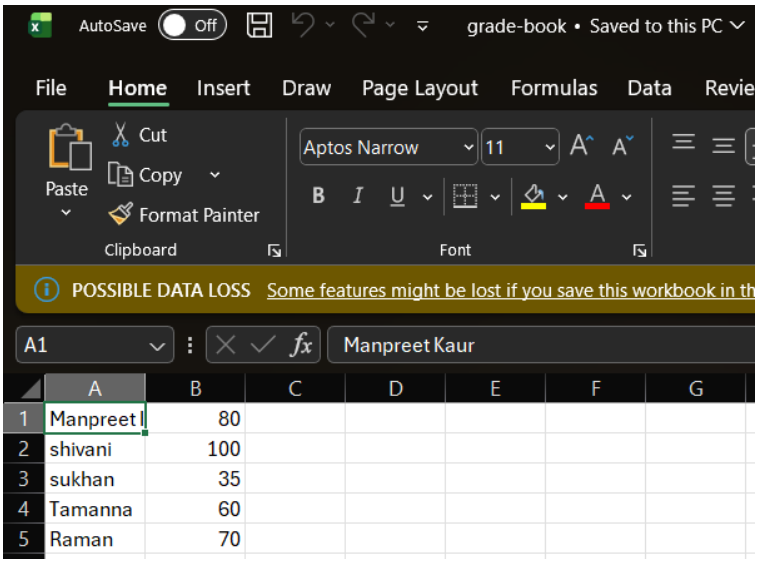
GRADE-BOOK-CSV.PY

C:\Users\MANPREET KAUR\Desktop\PYTHON\Grade-Book-Analyzer\grade-book-csv.py

grade-book-csv.py × Grade-Book-Analyzer.py* ×

```
1  import csv
2  with open("grade-book.csv","w",newline="") as f:
3      writer=csv.writer(f)
4      writer.writerow(["Manpreet Kaur",80])
5      writer.writerow(["shivani",100])
6      writer.writerow(["sukhan",35])
7      writer.writerow(["Tamanna",60])
8      writer.writerow(["Raman",70])
```

GRADE-BOOK-CSV.CSV



OUTPUT

```
Console 1/A
In [11]: %runfile 'C:/Users/MANPREET KAUR/Desktop/PYTHON/Grade-Book-Analyzer/Grade-Book-Analyzer.py' --wdir
Welcome to Grade-Book-Analyzer !
Choose data input method :
1. Manual input
2. Load from a CSV file
Enter your choice (1 or 2) : 2
Enter CSV filename : grade-book.csv
Statistics Summary :
Average marks : 69.0
Median marks : 70
Max Score : 100 by shivani
Mix Score : 35 by sukhan
Grade Distribution :
A : 1 students
B : 1 students
C : 1 students
D : 1 students
F : 1 students
Passed Students (4) : Manpreet Kaur,shivani,Tamanna,Raman
Failed students (1) : sukhan
```

```
Console 1/A
Passed Students (4) : Manpreet Kaur,shivani,Tamanna,Raman
Failed students (1) : sukhan
Name            Marks    Grade
-------------------------------------
Manpreet Kaur      80         B
shivani           100         A
sukhan             35         F
Tamanna            60         D
Raman              70         C
-------------------------------------
Do you want to analyze another set? (yes/no) : yes
Welcome to Grade-Book-Analyzer !
Choose data input method :
1. Manual input
2. Load from a CSV file
Enter your choice (1 or 2) : 1
Enter the number of students : 5
Enter the name of the student : Manpreet
Enter the marks of Manpreet : 99
Enter the name of the student : Raman
Enter the marks of Raman : 34
Enter the name of the student : Shivani
Enter the marks of Shivani : 50
```

```
Enter the name of the student : Sukhan
Enter the marks of Sukhan : 70
Enter the name of the student : Khushi
Enter the marks of Khushi : 79
Statistics Summary :
Average marks : 66.4
Median marks : 70
Max Score : 99 by Manpreet
Mix Score : 34 by Raman
Grade Distribution :
A : 1 students
B : 0 students
C : 2 students
D : 0 students
F : 2 students
Passed Students (4) : Manpreet,Shivani,Sukhan,Khushi
Failed students (1) : Raman
```

```
Name                Marks    Grade
-------------------------------------
Manpreet              99         A
Raman                 34         F
Shivani               50         F
Sukhan                70         C
Khushi                79         C
-------------------------------------
Do you want to analyze another set? (yes/no) : no
Thank you for using Grade-Book-Analyzer !
```

THANK YOU !