Lab #1: GPA & CGPA Calculator - Solutions

■ Solution in Python

1. **Single Student GPA Calculator**
```python
def marks_to_gradepoint(marks):
if marks >= 85: return 4.0
elif marks >= 80: return 3.7
elif marks >= 75: return 3.3
elif marks >= 70: return 3.0
elif marks >= 65: return 2.7
elif marks >= 60: return 2.3
elif marks >= 55: return 2.0
elif marks >= 50: return 1.7
else: return 0.0

def calculate_gpa(marks_list, credit_hours):
total_points = sum(marks_to_gradepoint(m) * c for m, c in zip(marks_list, credit_hours))
total_credits = sum(credit_hours)
return round(total_points / total_credits, 2)

marks = [88, 72, 67]
credits = [3, 3, 2]
print("GPA:", calculate_gpa(marks, credits))
```

2. **GPA/CGPA from Excel**
```python
import pandas as pd

def marks_to_gp(marks):
if marks >= 85: return 4.0
elif marks >= 80: return 3.7
elif marks >= 75: return 3.3
elif marks >= 70: return 3.0
elif marks >= 65: return 2.7
elif marks >= 60: return 2.3
elif marks >= 55: return 2.0
elif marks >= 50: return 1.7
else: return 0.0

df = pd.read_excel("students_marks.xlsx")
df["GradePoint"] = df["Marks"].apply(marks_to_gp)
df["WeightedGP"] = df["GradePoint"] * df["CreditHours"]

gpa = df.groupby("StudentID").apply(lambda x: x["WeightedGP"].sum() / x["CreditHours"].sum())
gpa = gpa.reset_index(name="GPA")

df.to_excel("students_gpa_output.xlsx", index=False)
print(gpa)
```

■ Solution in Excel

1. Use `VLOOKUP` or `IFS` for grade point mapping.
Example:
=IFS(A2>=85,4.0, A2>=80,3.7, A2>=75,3.3, A2>=70,3.0, A2>=65,2.7, A2>=60,2.3, A2>=55,2.0, A2>=50,1.7, A2<50,0)

2. Calculate Weighted Grade Points:
= GradePoint * CreditHours

3. GPA for semester:
= SUM(WeightedGP) / SUM(CreditHours)

4. CGPA across semesters:
= SUM(All WeightedGP) / SUM(All CreditHours)

■ Bonus Task Solution: GPA Prediction

- In Python: Prompt user for expected marks, recalculate GPA with the same formula.
- In Excel: Add a column 'Expected Marks' and apply the same IFS() formula to calculate Predicted GPA.