

Task 3.2C Process data using Pandas

- ✓ You are given a student result data file (result_withoutTotal.csv). It has columns: ID: student id Ass1 ~ Ass4: assignment scores (out of 100); weight of ass1, ass2, ass3 and ass4 is 5%, 15%, 5%, and 15%, respectively. Exam: examination score (out of 120); weight is 60%. Total score can be calculated using formula: $\text{Total} = 5\% * (\text{ass1} + \text{ass3}) + 15\% * (\text{ass2} + \text{ass4}) + 50\% * \text{exam}$ (as exam is out of 120) Read students' result data from file result_withoutTotal.csv,
 - Total column: $\text{Total} = 5\% * (\text{ass1} + \text{ass3}) + 15\% * (\text{ass2} + \text{ass4}) + 50\% * \text{exam}$.
 - Final column: Final = Total score rounded to the nearest integer.
 - To pass the unit, a student must achieve at least 50 of the Total and 40% of Exam which is 48 out of 120 (or $\text{Total} \geq 50$ and $\text{Exam} \geq 48$).
 - If a student failed the hurdle ($\text{Exam} \geq 48$), the max Final is 44. No change to Final score if Final
 - Grade column: N ($\text{Final} \leq 49.45$), P ($49.45 < \text{Final} \leq 59.45$), C ($59.45 < \text{Final} \leq 69.45$), D ($69.45 < \text{Final} \leq 79.45$) and HD ($79.45 < \text{Final}$). Border values are as follows:

Save

- the result data file with the 3 new columns to a file called result_updated.csv.
- the students' records with exam score < 48 to a file called failedhurdle.csv

Display

- the result data file with the 3 new columns
 - the students with exam score < 48 (these who failed the hurdle)
 - the students with exam score > 100
-
- This Python script processes student results from a CSV file to calculate total and final scores, apply a hurdle rule, and assign grades. It uses Pandas to read the CSV file and compute the total score based on assignment and exam weights, then rounds it to create the final score. The script enforces a hurdle rule where students must score at least 50 overall and 48 in the exam to pass, capping the final score at 44 for those who fail the exam. It assigns grades based on the final score and saves the updated data to a new CSV file. Additionally, it saves and displays records of students failing the hurdle and those with exam scores above 100.

```

jupyter 1806 S 2 Last checkpoint a minute ago (auto-saved)
File Edit View Insert Cell Format Help Python 3 (system)

In [18]: Import pandas as pd # Importing the pandas library for data manipulation

# Read the CSV file
df = pd.read_csv("C:/Users/robert/Desktop/2nd year/CS analysis/Week 01/3.3C/result_albmodTotal.csv")

# Calculate the total score using the given formula
df['Total'] = 0.05 * (df['Ass1'] + df['Ass2']) + 0.15 * (df['Ass3'] + df['Ass4']) + 0.5 * df['Exam']

# Calculate the final score by rounding the Total score to the nearest integer
df['Final'] = df['Total'].round()

# Apply the hurdle rule
# Students must meet a Total >= 60 and Exam >= 40 to pass the unit
# If a student fails the hurdle (Exam < 40), the new Final is 40
df.loc[df['Exam'] < 40, 'Final'] = df.loc[df['Exam'] < 40, 'Final'].apply(lambda x: min(x, 40))

# Assign grade based on the Final score
def assign_grade(final_score):
    if final_score <= 49.99:
        return 'F'
    elif final_score <= 59.99:
        return 'D'
    elif final_score <= 69.99:
        return 'C'
    elif final_score <= 79.99:
        return 'B'
    else:
        return 'A'

df['Grade'] = df['Final'].apply(assign_grade)

# Save the result data with the new columns to a new file
df.to_csv("C:/Users/robert/Desktop/2nd year/CS analysis/Week 01/3.3C/result_updated.csv", index=False)

# Save the records of students with exam score < 40 to a new file
failed_hurdle = df[df['Exam'] < 40]
failed_hurdle.to_csv("C:/Users/robert/Desktop/2nd year/CS analysis/Week 01/3.3C/failedhurdle.csv", index=False)

# Display the updated result data
print("Updated result data with Total, Final, and Grade columns:")
print(df.to_string(index=False)) # Displaying without the index for a cleaner look

# Display the students who failed the hurdle (Exam < 40)
print("\nStudents who failed the hurdle (Exam < 40):")
print(failed_hurdle.to_string(index=False)) # Displaying without the index for a cleaner look

# Display the students with exam score < 60
print("\nStudents with exam score < 60")
print(df[df['Exam'] < 60].to_string(index=False)) # Displaying without the index for a cleaner look

```

Updated result data with Total, Final, and Grade columns:

ID	Ass1	Ass2	Ass3	Ass4	Exam	Total	Final	Grade	total
1	89.1	50.0	85.0	88.9	65	62.040	62.0	C	68.540
2	95.1	82.5	90.5	94.5	52	61.830	62.0	C	67.030
3	74.3	54.4	63.0	63.9	31	40.110	40.0	N	43.210
4	89.8	81.3	82.0	90.4	37	52.845	44.0	N	56.545
5	91.3	98.8	92.5	95.9	79	77.895	78.0	D	85.795
6	83.9	82.5	89.0	98.6	68	69.810	70.0	D	76.610
7	81.9	50.0	68.5	95.4	59	58.830	59.0	P	64.730
8	50.0	54.9	50.0	87.7	51	51.890	52.0	P	56.990
9	90.5	65.9	50.0	72.2	63	59.240	59.0	P	65.540
10	89.0	89.9	94.0	90.3	84	78.180	78.0	D	86.580
11	96.6	100.0	98.0	97.3	102	90.325	90.0	HD	100.525
12	58.8	67.5	72.0	75.4	51	53.475	53.0	P	58.575
13	74.7	78.5	50.0	86.2	66	63.940	64.0	C	70.540
14	87.7	80.0	57.0	73.6	52	56.275	56.0	P	61.475
15	66.3	53.7	53.0	81.9	30	41.305	41.0	N	44.305
16	82.7	77.2	73.0	95.9	50	58.750	59.0	P	63.750
17	97.5	96.2	83.5	93.8	55	65.050	65.0	C	70.550
18	70.0	60.0	57.0	70.0	50	56.625	57.0	D	60.625

ID	Ass1	Ass2	Ass3	Ass4	Exam	Total	Final	Grade	total
3	74.3	54.4	63.0	63.9	31	40.110	40.0	N	43.210
4	89.8	81.3	82.0	90.4	37	52.845	44.0	N	56.545
15	66.3	53.7	53.0	81.9	30	41.305	41.0	N	44.305
24	57.7	76.3	71.0	87.7	35	48.535	44.0	N	52.035
25	84.7	65.0	73.0	88.9	34	47.970	44.0	N	51.370
26	84.7	53.8	75.0	78.1	36	45.770	44.0	N	49.370
33	64.2	50.0	18.0	0.0	0	11.610	12.0	N	11.610
42	81.5	43.8	0.0	0.0	0	10.645	11.0	N	10.645
44	71.9	61.3	76.0	94.5	38	49.765	44.0	N	53.565
47	50.0	71.3	56.0	93.8	34	47.065	44.0	N	50.465
54	76.1	50.0	50.0	50.0	33	37.805	38.0	N	41.105
60	73.9	53.2	74.0	95.9	34	46.760	44.0	N	50.160
78	52.9	53.2	50.0	50.0	36	38.625	39.0	N	42.225