

Programming Languages and Compilers

Lectured by Prof. Chung Yung

Programming Assignment 2

Clara Gabriella Choc – 411021326
Samuel Ian Chavarria - 411021334

1. Problem description

Practice on programming the same exercises in 5 different programming languages with Java, Python, R, ML, and Prolog. The grade of a student is translated according to the following table:

Score	Grade
0-49	E
50-59	D
60-62	C-
63-66	C
67-69	C+
70-72	B-
73-76	B
77-79	B+
80-84	A-
85-89	A
90-100	A+

The overall score of a student is calculated by the following formula:

$$\text{Score} = \text{HW1} * 0.1 + \text{HW2} * 0.1 + \text{HW3} * 0.1 + \text{Midterm} * 0.3 + \text{Final} * 0.4$$

Write a program in the above 5 different programming languages to translate the overall score of each student into a grade. Round off the overall scores into integers before looking up the table. Note that for Java, Python, and R, it is required to read the grade standards from HW2grade.csv, while for ML and Prolog, it is allowed to give the grade standards in the program.

2. Highlight The Way You Write The Program

Python

- We first read the CSV file to our program and then store it in data frame.
- Operate the data frame to get each score (HW, midterm and final) of all students
- Calculate the average score based on the given formula and add it to a list
- Translated the calculated score to the corresponding grade and add it to another list
- Add the score and grade list to the end of the original data frame
- Print out the data frame

R

- We first read the CSV file to our program and store it in a 2D-array
- Calculate the average score based on the given formula and add it to an array named score
- Go through the score array and find out the corresponding grade, then add it to another array named grade
- Add the original 2D-array, score array and grade array to data frame
- Print out the data frame

Java

- Use File to read the CSV file
- Create a 2D-list to store the CSV file
- Declare an array list named score to store the calculated score based on the given formula
- Declare another array list named grade to store the corresponding grade according to the score
- Print out the 2D-list (original data include name, id, HW, mid and final), score array list and grade array list

ML

- Declare the name, id, hw1-3, midterm and final list in the program
- For list hw1 ~ hw3, * 0.1, for the midterm list, *0.3 and final list *0.4
- Add the above list together so we can get the score for each student
- Using a function to go through the score list and store the corresponding grade to each independent variable
- Store each student's name, id, score in different variable
- Print out those variable(adjust it to print out like a data frame)

Prolog

- Declare arrays for each students, which contains NO, id, name and score for HW1-3, midterm and final.
 - For each array, calculate the score based on the given formula and store it in an independent variable
 - Write a function for printing out.
 - While printing out the original data, convert each student's score to the corresponding grade and then also print it out.
3. Program Listing

3. Program Listing

Python

```
1 import pandas as pd
2
3 data = pd.read_csv("HW2data.csv")
4 scorearr = []
5 gradearr = []
6 for i in range(len(data)):
7     score = (int(data.loc[i,"HW1"])+ int(data.loc[i,"HW2"]) + int(data.loc[i,"HW3"]))*0.1 + int(data.loc[i,"Midterm"])*0.3 + int(data.loc[i,"Final"])*0.4
8     scorearr.append(score)
9     if score >= 89.5:
10         grade = "A+"
11     elif score >= 84.5:
12         grade = "A"
13     elif score >= 79.5:
14         grade = "A-"
15     elif score >= 76.5:
16         grade = "B+"
17     elif score >= 72.5:
18         grade = "B"
19     elif score >= 69.5:
20         grade = "B-"
21     elif score >= 66.5:
22         grade = "C+"
23     elif score >= 62.5:
24         grade = "C"
25     elif score >= 59.5:
26         grade = "C-"
27     elif score >= 49.5:
28         grade = "D"
29     else:
30         grade = "E"
31     gradearr.append(grade)
32
33 data['Score'] = scorearr
34 data['Grade'] = gradearr
35 print(data)
```

Java

Prolog

```
1 :- initialization(main).
2
3 print(No,Id,Name,S,Hw1,Hw2,Hw3,Mid,Final) :-
4 ( S >= 89.5 ->
5   write(No),write(Id),write(Name),write(Hw1),write(Hw2),write(Hw3),write(Mid),write(Final),format(' ~2f', [S]),write(' A+');
6   S >= 84.5 ->
7   write(No),write(Id),write(Name),write(Hw1),write(Hw2),write(Hw3),write(Mid),write(Final),format(' ~2f', [S]),write(' A');
8   S >= 79.5 ->
9   write(No),write(Id),write(Name),write(Hw1),write(Hw2),write(Hw3),write(Mid),write(Final),format(' ~2f', [S]),write(' A-');
10  S >= 76.5 ->
11  write(No),write(Id),write(Name),write(Hw1),write(Hw2),write(Hw3),write(Mid),write(Final),format(' ~2f', [S]),write(' B+');
12  S >= 72.5 ->
13  write(No),write(Id),write(Name),write(Hw1),write(Hw2),write(Hw3),write(Mid),write(Final),format(' ~2f', [S]),write(' B');
14  S >= 69.5 ->
15  write(No),write(Id),write(Name),write(Hw1),write(Hw2),write(Hw3),write(Mid),write(Final),format(' ~2f', [S]),write(' B-');
16  S >= 66.5 ->
17  write(No),write(Id),write(Name),write(Hw1),write(Hw2),write(Hw3),write(Mid),write(Final),format(' ~2f', [S]),write(' C+');
18  S >= 62.5 ->
19  write(No),write(Id),write(Name),write(Hw1),write(Hw2),write(Hw3),write(Mid),write(Final),format(' ~2f', [S]),write(' C');
20  S >= 59.5 ->
21  write(No),write(Id),write(Name),write(Hw1),write(Hw2),write(Hw3),write(Mid),write(Final),format(' ~2f', [S]),write(' C-');
22  S >= 49.5 ->
23  write(No),write(Id),write(Name),write(Hw1),write(Hw2),write(Hw3),write(Mid),write(Final),format(' ~2f', [S]),write(' D');
24  write(No),write(Id),write(Name),write(Hw1),write(Hw2),write(Hw3),write(Mid),write(Final),format(' ~2f', [S]),write(' E')
25 ).
26
27 main:- Alan = [' 1 ', '410021001.', Alan',90,84.5,117,60,66 ],
28 Bob = [' 2 ', '410021002.', Bob',85,49,98,57,64 ],
29 Carrie = [' 3 ', '410021003.', Carrie',90,110.5,117,60,62 ],
30 David = [' 4 ', '410021004.', David',117,85,0,44,55 ],
31 Ethan = [' 5 ', '410021005.', Ethan',85,56,50,57,67 ],
32 Frank = [' 6 ', '410021006.', Frank',90,65,65,72,66 ],
33 Gary = [' 7 ', '410021007.', Gary',117,110.5,65,69,43 ],
34 Helen = [' 8 ', '410021008.', Helen',117,65,50,43,54 ],
35 Igor = [' 9 ', '410021009.', Igor',63,59.5,50,51,75 ],
36 Jeff = [' 10 ', '410021010.', Jeff',117,110.5,117,53,75 ],
37 King = [' 11 ', '410021011.', King',100,100,100,56,60 ],
38 Leo = [' 12 ', '410021012.', Leo',95,90,85,58,53 ],
39 Mark = [' 13 ', '410021013.', Mark',100,90,90,81,72 ],
40 Nancy = [' 14 ', '410021014.', Nancy',80,85,90,64,57 ],
41 Oliver = [' 15 ', '410021015.', Oliver',75,90,100,72,63 ],
42 Peter = [' 16 ', '410021016.', Peter',60,75,85,69,58 ],
43 Quincy = [' 17 ', '410021017.', Quincy',85,80,90,44,39 ],
44 Richar = [' 18 ', '410021018.', Richar',90,85,80,52,48 ],
45 Steve = [' 19 ', '410021019.', Steve',100,90,90,83,61 ],
46 Tom = [' 20 ', '410021020.', Tom',100,100,100,76,74 ],
47
48 nth0(3,Alan,A1),nth0(4,Alan,A2),nth0(5,Alan,A3),nth0(6,Alan,Amid),nth0(7,Alan,Afinal),
49 nth0(0,Alan,A0),nth0(1,Alan,Aid),nth0(2,Alan,Aname),
50 A is A1*0.1+A2*0.1+A3*0.1+Amid*0.3+Afinal*0.4,
51 nth0(3,Bob,B1),nth0(4,Bob,B2),nth0(5,Bob,B3),nth0(6,Bob,Bmid),nth0(7,Bob,Bfinal),
52 nth0(0,Bob,B0),nth0(1,Bob,Bid),nth0(2,Bob,Bname),
53 B is B1*0.1+B2*0.1+B3*0.1+Mid*0.3+Bfinal*0.4,
54 nth0(3,Carrie,C1),nth0(4,Carrie,C2),nth0(5,Carrie,C3),nth0(6,Carrie,Cmid),nth0(7,Carrie,Cfinal),
55 nth0(0,Carrie,C0),nth0(1,Carrie,Cid),nth0(2,Carrie,Cname),
56 C is C1*0.1+C2*0.1+C3*0.1+Cmid*0.3+Cfinal*0.4,
57 nth0(3,David,D1),nth0(4,David,D2),nth0(5,David,D3),nth0(6,David,Dmid),nth0(7,David,Dfinal),
58 nth0(0,David,D0),nth0(1,David,Did),nth0(2,David,Dname),
59 D is D1*0.1+D2*0.1+D3*0.1+Dmid*0.3+Dfinal*0.4,
60 nth0(3,Ethan,E1),nth0(4,Ethan,E2),nth0(5,Ethan,E3),nth0(6,Ethan,Emid),nth0(7,Ethan,Efinal),
61 nth0(0,Ethan,E0),nth0(1,Ethan,Eid),nth0(2,Ethan,Ename),
62 E is E1*0.1+E2*0.1+E3*0.1+Emid*0.3+Efinal*0.4,
63 nth0(3,Frank,F1),nth0(4,Frank,F2),nth0(5,Frank,F3),nth0(6,Frank,Fmid),nth0(7,Frank,Ffinal),
64 nth0(0,Frank,F0),nth0(1,Frank,Fid),nth0(2,Frank,Fname),
65 F is F1*0.1+F2*0.1+F3*0.1+Fmid*0.3+Ffinal*0.4,
66 nth0(3,Gary,G1),nth0(4,Gary,G2),nth0(5,Gary,G3),nth0(6,Gary,Gmid),nth0(7,Gary,Gfinal),
67 nth0(0,Gary,G0),nth0(1,Gary,Gid),nth0(2,Gary,Gname),
68 G is G1*0.1+G2*0.1+G3*0.1+Gmid*0.3+Gfinal*0.4,
69 nth0(3,Helen,H1),nth0(4,Helen,H2),nth0(5,Helen,H3),nth0(6,Helen,Hmid),nth0(7,Helen,Hfinal),
70 nth0(0,Helen,H0),nth0(1,Helen,Hid),nth0(2,Helen,Hname),
71 H is H1*0.1+H2*0.1+H3*0.1+Hmid*0.3+Hfinal*0.4,
72 nth0(3,Igor,I1),nth0(4,Igor,I2),nth0(5,Igor,I3),nth0(6,Igor,Imid),nth0(7,Igor,Ifinal),
73 nth0(0,Igor,I0),nth0(1,Igor,Iid),nth0(2,Igor,Iname),
74 I is I1*0.1+I2*0.1+I3*0.1+Imid*0.3+Ifinal*0.4,
75 nth0(3,Jeff,J1),nth0(4,Jeff,J2),nth0(5,Jeff,J3),nth0(6,Jeff,Jmid),nth0(7,Jeff,Jfinal),
76 nth0(0,Jeff,J0),nth0(1,Jeff,Jid),nth0(2,Jeff,Jname),
77 J is J1*0.1+J2*0.1+J3*0.1+Jmid*0.3+Jfinal*0.4,
78 nth0(3,King,K1),nth0(4,King,K2),nth0(5,King,K3),nth0(6,King,Kmid),nth0(7,King,Kfinal),
79 nth0(0,King,K0),nth0(1,King,Kid),nth0(2,King,Kname),
80 K is K1*0.1+K2*0.1+K3*0.1+Kmid*0.3+Kfinal*0.4,
81 nth0(3,Leo,L1),nth0(4,Leo,L2),nth0(5,Leo,L3),nth0(6,Leo,Lmid),nth0(7,Leo,Lfinal),
82 nth0(0,Leo,L0),nth0(1,Leo,Lid),nth0(2,Leo,Lname),
83 L is L1*0.1+L2*0.1+L3*0.1+Lmid*0.3+Lfinal*0.4,
84 nth0(3,Mark,M1),nth0(4,Mark,M2),nth0(5,Mark,M3),nth0(6,Mark,Mmid),nth0(7,Mark,Mfinal),
85 nth0(0,Mark,M0),nth0(1,Mark,Mid),nth0(2,Mark,Mname),
86 M is M1*0.1+M2*0.1+M3*0.1+Mmid*0.3+Mfinal*0.4,
87 nth0(3,Nancy,N1),nth0(4,Nancy,N2),nth0(5,Nancy,N3),nth0(6,Nancy,Nmid),nth0(7,Nancy,Nfinal),
88 nth0(0,Nancy,N0),nth0(1,Nancy,Nid),nth0(2,Nancy,Nname),
89 N is N1*0.1+N2*0.1+N3*0.1+Nmid*0.3+Nfinal*0.4,
90 nth0(3,Oliver,O1),nth0(4,Oliver,O2),nth0(5,Oliver,O3),nth0(6,Oliver,Omid),nth0(7,Oliver,Ofinal),
91 nth0(0,Oliver,O0),nth0(1,Oliver,Oid),nth0(2,Oliver,Oname),
92 O is O1*0.1+O2*0.1+O3*0.1+Omid*0.3+Ofinal*0.4,
93 nth0(3,Peter,P1),nth0(4,Peter,P2),nth0(5,Peter,P3),nth0(6,Peter,Pmid),nth0(7,Peter,Pfinal),
94 nth0(0,Peter,P0),nth0(1,Peter,Pid),nth0(2,Peter,Pname),
95 P is P1*0.1+P2*0.1+P3*0.1+Pmid*0.3+Pfinal*0.4,
96 nth0(3,Quincy,Q1),nth0(4,Quincy,Q2),nth0(5,Quincy,Q3),nth0(6,Quincy,Qmid),nth0(7,Quincy,Qfinal),
97 nth0(0,Quincy,Q0),nth0(1,Quincy,Qid),nth0(2,Quincy,Qname),
98 Q is Q1*0.1+Q2*0.1+Q3*0.1+Qmid*0.3+Qfinal*0.4,
99 nth0(3,Richar,R1),nth0(4,Richar,R2),nth0(5,Richar,R3),nth0(6,Richar,Rmid),nth0(7,Richar,Rfinal),
100 nth0(0,Richar,R0),nth0(1,Richar,Rid),nth0(2,Richar,Rname),
101 R is R1*0.1+R2*0.1+R3*0.1+Rmid*0.3+Rfinal*0.4,
102 nth0(3,Steve,S1),nth0(4,Steve,S2),nth0(5,Steve,S3),nth0(6,Steve,Smid),nth0(7,Steve,Sfinal),
103 nth0(0,Steve,S0),nth0(1,Steve,Sid),nth0(2,Steve,Sname),
104 S is S1*0.1+S2*0.1+S3*0.1+Smid*0.3+Sfinal*0.4,
105 nth0(3,Tom,T1),nth0(4,Tom,T2),nth0(5,Tom,T3),nth0(6,Tom,Tmid),nth0(7,Tom,Tfinal),
106 nth0(0,Tom,T0),nth0(1,Tom,Tid),nth0(2,Tom,Tname),
107 T is T1*0.1+T2*0.1+T3*0.1+Tmid*0.3+Tfinal*0.4,
108
109
110 write('NO ID NAME HW1 HW2 HW3 Midterm Final Score Grade'),nl,
111 print(Alan,Aid,Aname,A, 90, 84.5, 117, 60, 66),nl,
112 print(Bob,Bid,Bname,B, 85, 49, 98, 57, 64),nl,
113 print(Cno,Cid,Cname,C, 90, 110.5, 117, 68, 62),nl,
114 print(Dno,Did,Dname,D, 117, 85, 0, 44, 55),nl,
115 print(Eno,Eid,Ename,E, 85, 56, 50, 57, 67),nl,
116 print(Fno,Fid,Fname,F, 90, 65, 65, 72, 66),nl,
117 print(Gno,Gid,Gname,G, 117, 110.5, 65, 69, 43),nl,
118 print(Hno,Hid,Hname,H, 117, 65, 50, 43, 54),nl,
119 print(Ino,Iid,Iname,I, 63, 59.5, 50, 51, 75),nl,
120 print(Jno,Jid,Jname,J, 117, 110.5, 117, 53, 75),nl,
121 print(Kno,Kid,Kname,K, 100, 100, 100, 56, 60),nl,
122 print(Lno,Lid,Lname,L, 95, 90, 85, 58, 53),nl,
123 print(Mno,Mid,Mname,M, 100, 90, 90, 81, 72),nl,
124 print(Nno,Nid,Nname,N, 80, 85, 90, 64, 57),nl,
125 print(Ono,Oid,Oname,O, 75, 90, 100, 72, 63),nl,
126 print(Pno,Pid,Pname,P, 60, 75, 85, 69, 58),nl,
127 print(Qno,Qid,Qname,Q, 85, 80, 90, 44, 39),nl,
128 print(Rno,Rid,Rname,R, 90, 85, 80, 52, 48),nl,
129 print(Sno,Sid,Sname,S, 100, 90, 90, 83, 61),nl,
130 print(Tno,Tid,Tname,T, 100, 100, 100, 76, 74).
```

ML

```

1 fun add_lists any [] = any
2   add_lists [] any = any
3   add_lists (a::bs) (c::ds) = real list = (a+c) :: add_lists bs ds;
4
5 fun gradeval (score: real) =
6   if score > 89.50 then "A+"
7   else if score > 84.50 then "A"
8   else if score > 79.50 then "A-"
9   else if score > 74.50 then "B+"
10  else if score > 69.50 then "B"
11  else if score > 64.50 then "B-"
12  else if score > 59.50 then "C+"
13  else if score > 54.50 then "C"
14  else if score > 49.50 then "C-"
15  else if score > 44.50 then "D"
16  else "F";
17
18 val no = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20] : int list;
19 val ID = ["410021001","410021002","410021003","410021004","410021005","410021006","410021007","410021008","410021009","410021010","410021011","410021012","410021013","410021014","410021015","410021016","410021017","410021018","410021019","410021020"] : string list;
20 val HW1 = [100.0,85.0,90.0,90.0,117.0,85.0,90.0,117.0,85.0,90.0,117.0,85.0,90.0,100.0,80.0,75.0,60.0,85.0,90.0,100.0] : real list;
21 val HW2 = [104.5,48.0,110.5,85.0,56.0,65.0,110.5,85.0,59.5,110.5,100.0,90.0,90.0,85.0,90.0,75.0,80.0,85.0,90.0,100.0] : real list;
22 val HW3 = [117.0,80.0,117.0,80.0,50.0,65.0,65.0,50.0,117.0,100.0,85.0,90.0,90.0,100.0,85.0,90.0,80.0,90.0,90.0,100.0] : real list;
23 val Midterm = [60.0,57.0,68.0,44.0,57.0,72.0,69.0,43.0,51.0,53.0,56.0,58.0,81.0,64.0,72.0,69.0,44.0,52.0,83.0,76.0] : real list;
24 val Final = [66.0,64.0,62.0,55.0,67.0,66.0,43.0,54.0,75.0,68.0,53.0,72.0,57.0,63.0,50.0,39.0,48.0,61.0,74.0] : real list;
25 val Name = ["Alan","Bob","Carrie","David","Ethan","Frank","Gary","Heleen","Igor","Jeff","King","Leo","Mark","Nancy","Oliver","Peter","Quincy","Richard","Steve","Tom"] : string list;
26
27 val HW1 = map(fn x => x*0.10) HW1;
28 val HW2 = map(fn x => x*0.10) HW2;
29 val HW3 = map(fn x => x*0.10) HW3;
30 val Mid = map(fn x => x*0.50) Midterm;
31 val Fin = map(fn x => x*0.40) Final;
32
33 val hw12 = add_lists HW1 HW2;
34 val hw123 = add_lists hw12 HW3;
35 val hw123mid = add_lists hw123 Mid;
36 val hw123mid final = add_lists hw123mid Fin : real list;
37
38 val (no1, no, std1, id1, sidname1, hw123 mid final, ID, Name) =
39   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
40 val grade1 = gradeval std1;
41
42 val (no2, no, std2, id2, sidname2, hw123 mid final, ID, Name) =
43   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
44 val grade2 = gradeval std2;
45
46 val (no3, no, std3, id3, sidname3, hw123 mid final, ID, Name) =
47   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
48 val grade3 = gradeval std3;
49
50 val (no4, no, std4, id4, sidname4, hw123 mid final, ID, Name) =
51   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
52 val grade4 = gradeval std4;
53
54 val (no5, no, std5, id5, sidname5, hw123 mid final, ID, Name) =
55   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
56 val grade5 = gradeval std5;
57
58 val (no6, no, std6, id6, sidname6, hw123 mid final, ID, Name) =
59   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
60 val grade6 = gradeval std6;
61
62 val (no7, no, std7, id7, sidname7, hw123 mid final, ID, Name) =
63   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
64 val grade7 = gradeval std7;
65
66 val (no8, no, std8, id8, sidname8, hw123 mid final, ID, Name) =
67   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
68 val grade8 = gradeval std8;
69
70 val (no9, no, std9, id9, sidname9, hw123 mid final, ID, Name) =
71   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
72 val grade9 = gradeval std9;
73
74 val (no10, no, std10, id10, sidname10, hw123 mid final, ID, Name) =
75   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
76 val grade10 = gradeval std10;
77
78 val (no11, no, std11, id11, sidname11, hw123 mid final, ID, Name) =
79   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
80 val grade11 = gradeval std11;
81
82 val (no12, no, std12, id12, sidname12, hw123 mid final, ID, Name) =
83   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
84 val grade12 = gradeval std12;
85
86 val (no13, no, std13, id13, sidname13, hw123 mid final, ID, Name) =
87   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
88 val grade13 = gradeval std13;
89
90 val (no14, no, std14, id14, sidname14, hw123 mid final, ID, Name) =
91   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
92 val grade14 = gradeval std14;
93
94 val (no15, no, std15, id15, sidname15, hw123 mid final, ID, Name) =
95   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
96 val grade15 = gradeval std15;
97
98 val (no16, no, std16, id16, sidname16, hw123 mid final, ID, Name) =
99   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
100 val grade16 = gradeval std16;
101
102 val (no17, no, std17, id17, sidname17, hw123 mid final, ID, Name) =
103   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
104 val grade17 = gradeval std17;
105
106 val (no18, no, std18, id18, sidname18, hw123 mid final, ID, Name) =
107   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
108 val grade18 = gradeval std18;
109
110 val (no19, no, std19, id19, sidname19, hw123 mid final, ID, Name) =
111   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
112 val grade19 = gradeval std19;
113
114 val (no20, no, std20, id20, sidname20, hw123 mid final, ID, Name) =
115   (hd(no), tl(no), hd(hw123 mid final), hd(ID), hd(Name), tl(hw123 mid final), tl(ID), tl(Name));
116 val grade20 = gradeval std20;
117
118 val = print "\n\n" "ID Name HW1 HW2 Midterm Final Score GradeVal" ^
119   Int.toString no1 ^ " " ^ ID1 ^ " " ^ sidname1 ^ " " ^ HW1 ^ " " ^ HW2 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
120 Int.toString no2 ^ " " ^ ID2 ^ " " ^ sidname2 ^ " " ^ HW2 ^ " " ^ HW3 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
121 Int.toString no3 ^ " " ^ ID3 ^ " " ^ sidname3 ^ " " ^ HW3 ^ " " ^ HW1 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
122 Int.toString no4 ^ " " ^ ID4 ^ " " ^ sidname4 ^ " " ^ HW1 ^ " " ^ HW2 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
123 Int.toString no5 ^ " " ^ ID5 ^ " " ^ sidname5 ^ " " ^ HW2 ^ " " ^ HW3 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
124 Int.toString no6 ^ " " ^ ID6 ^ " " ^ sidname6 ^ " " ^ HW3 ^ " " ^ HW1 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
125 Int.toString no7 ^ " " ^ ID7 ^ " " ^ sidname7 ^ " " ^ HW1 ^ " " ^ HW2 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
126 Int.toString no8 ^ " " ^ ID8 ^ " " ^ sidname8 ^ " " ^ HW2 ^ " " ^ HW3 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
127 Int.toString no9 ^ " " ^ ID9 ^ " " ^ sidname9 ^ " " ^ HW3 ^ " " ^ HW1 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
128 Int.toString no10 ^ " " ^ ID10 ^ " " ^ sidname10 ^ " " ^ HW1 ^ " " ^ HW2 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
129 Int.toString no11 ^ " " ^ ID11 ^ " " ^ sidname11 ^ " " ^ HW2 ^ " " ^ HW3 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
130 Int.toString no12 ^ " " ^ ID12 ^ " " ^ sidname12 ^ " " ^ HW3 ^ " " ^ HW1 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
131 Int.toString no13 ^ " " ^ ID13 ^ " " ^ sidname13 ^ " " ^ HW1 ^ " " ^ HW2 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
132 Int.toString no14 ^ " " ^ ID14 ^ " " ^ sidname14 ^ " " ^ HW2 ^ " " ^ HW3 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
133 Int.toString no15 ^ " " ^ ID15 ^ " " ^ sidname15 ^ " " ^ HW3 ^ " " ^ HW1 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
134 Int.toString no16 ^ " " ^ ID16 ^ " " ^ sidname16 ^ " " ^ HW1 ^ " " ^ HW2 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
135 Int.toString no17 ^ " " ^ ID17 ^ " " ^ sidname17 ^ " " ^ HW2 ^ " " ^ HW3 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
136 Int.toString no18 ^ " " ^ ID18 ^ " " ^ sidname18 ^ " " ^ HW3 ^ " " ^ HW1 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
137 Int.toString no19 ^ " " ^ ID19 ^ " " ^ sidname19 ^ " " ^ HW1 ^ " " ^ HW2 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";
138 Int.toString no20 ^ " " ^ ID20 ^ " " ^ sidname20 ^ " " ^ HW2 ^ " " ^ HW3 ^ " " ^ Midterm ^ " " ^ Final ^ " " ^ Score ^ " " ^ GradeVal ^ "\n\n";

```


R

```
1 # Read student scores file
2 setwd("/home/clara/Downloads/HW2_PL_Track/PL Track")
3 student_scores <- read.csv("HW2data.csv")
4
5 # Calculate overall scores
6 student_scores$Score <- (student_scores$HW1 * 0.1 + student_scores$HW2 * 0.1
7   + student_scores$HW3 * 0.1
8   + student_scores$Midterm * 0.3
9   + student_scores$Final * 0.4)
10
11 # Round off overall scores to 2 decimal places
12 student_scores$Score <- round(student_scores$Score, 2)
13
14 # Read grade standards file
15 grade_standards <- read.csv("HW2grade.csv")
16
17 # Translate overall scores to grades
18 translate_score_to_grade <- function(score) {
19   for (i in 1:nrow(grade_standards)) {
20     if (ceiling(score) >= grade_standards[i, "From"] && ceiling(score) <= grade_standards[i, "To"]) {
21       return(grade_standards[i, "Grade"])
22     }
23   }
24   return("")
25 }
26
27 student_scores$Grade <- sapply(student_scores$Score, translate_score_to_grade)
28
29 # Print the results
30 print(student_scores)
```

4. Test Results

Prolog

```
clara@clara-HP-Laptop-15:~/Downloads/HW2_PL_Track/PL_Track$ swipl -s prolog.pl
```

NO	ID	NAME	HW1	HW2	HW3	Midterm	Final	Score	Grade
1	410021001	Alan	90	84.5	117	60	66	73.55	B
2	410021002	Bob	85	49	80	57	64	64.10	C
3	410021003	Carrie	90	110.5	117	68	62	76.95	B+
4	410021004	David	117	85	0	44	55	55.40	D
5	410021005	Ethan	85	56	50	57	67	63.00	C
6	410021006	Frank	90	65	65	72	66	70.00	B-
7	410021007	Gary	117	110.5	65	69	43	67.15	C+
8	410021008	Helen	117	65	50	43	54	57.70	D
9	410021009	Igor	63	59.5	50	51	75	62.55	C
10	410021010	Jeff	117	110.5	117	53	75	80.35	A-
11	410021011	King	100	100.0	100	56	68	74.00	B
12	410021012	Leo	95	90	85	58	53	65.60	C
13	410021013	Mark	100	90	90	81	72	81.10	A-
14	410021014	Nancy	80	85	90	64	57	67.50	C+
15	410021015	Oliver	75	90	100	72	63	73.30	B
16	410021016	Peter	60	75	85	69	58	65.90	C
17	410021017	Quincy	85	80	90	44	39	54.30	D
18	410021018	Richar	90	85	80	52	48	60.30	C-
19	410021019	Steve	100	90	90	83	61	77.30	B+
20	410021020	Tom	100	100	100	76	74	82.40	A-

Java

```
clara@clara-HP-Laptop-15:~/Downloads/HW2_PL_Track/PL_Track$ java java
```

NO	ID	Name	HW1	HW2	HW3	Midterm	Final	Score	Grade
1	410021001	Alan	90	84.5	117	60	66	73.55	B
2	410021002	Bob	85	49	80	57	64	64.10	C
3	410021003	Carrie	90	110.5	117	68	62	76.95	B+
4	410021004	David	117	85	0	44	55	55.40	D
5	410021005	Ethan	85	56	50	57	67	63.00	C
6	410021006	Frank	90	65	65	72	66	70.00	B-
7	410021007	Gary	117	110.5	65	69	43	67.15	C+
8	410021008	Helen	117	65	50	43	54	57.70	D
9	410021009	Igor	63	59.5	50	51	75	62.55	C
10	410021010	Jeff	117	110.5	117	53	75	80.35	A-
11	410021011	King	100	100	100	56	68	74.00	B
12	410021012	Leo	95	90	85	58	53	65.60	C
13	410021013	Mark	100	90	90	81	72	81.10	A-
14	410021014	Nancy	80	85	90	64	57	67.50	C+
15	410021015	Oliver	75	90	100	72	63	73.30	B
16	410021016	Peter	60	75	85	69	58	65.90	C
17	410021017	Quincy	85	80	90	44	39	54.30	D
18	410021018	Richar	90	85	80	52	48	60.30	C-
19	410021019	Steve	100	90	90	83	61	77.30	B+
20	410021020	Tom	100	100	100	76	74	82.40	A-

Python

```
clara@clara-HP-Laptop-15:~/Downloads/HW2_PL_Track/PL_Track$ python3 python.py
```

	NO	ID	Name	HW1	HW2	HW3	Midterm	Final	Score	Grade
0	1	410021001	Alan	90	84.5	117	60	66	73.55	B
1	2	410021002	Bob	85	49.0	80	57	64	64.10	C
2	3	410021003	Carrie	90	110.5	117	68	62	76.95	B+
3	4	410021004	David	117	85.0	0	44	55	55.40	D
4	5	410021005	Ethan	85	56.0	50	57	67	63.00	C
5	6	410021006	Frank	90	65.0	65	72	66	70.00	B-
6	7	410021007	Gary	117	110.5	65	69	43	67.15	C+
7	8	410021008	Helen	117	65.0	50	43	54	57.70	D
8	9	410021009	Igor	63	59.5	50	51	75	62.55	C
9	10	410021010	Jeff	117	110.5	117	53	75	80.35	A-
10	11	410021011	King	100	100.0	100	56	68	74.00	B
11	12	410021012	Leo	95	90.0	85	58	53	65.60	C
12	13	410021013	Mark	100	90.0	90	81	72	81.10	A-
13	14	410021014	Nancy	80	85.0	90	64	57	67.50	C+
14	15	410021015	Oliver	75	90.0	100	72	63	73.30	B
15	16	410021016	Peter	60	75.0	85	69	58	65.90	C
16	17	410021017	Quincy	85	80.0	90	44	39	54.30	D
17	18	410021018	Richar	90	85.0	80	52	48	60.30	C-
18	19	410021019	Steve	100	90.0	90	83	61	77.30	B+
19	20	410021020	Tom	100	100.0	100	76	74	82.40	A-

ML

NO	ID	NAME	HW1	HW2	HW3	Midterm	Final	Score	Grade
1	410021001	Alan	90	84.5	117	60	66	73.55	B
2	410021002	Bob	85	49	80	57	64	64.1	C
3	410021003	Carrie	90	110.5	117	68	62	76.95	B+
4	410021004	David	117	85	0	44	55	55.4	D
5	410021005	Ethan	85	56	50	57	67	63.0	C
6	410021006	Frank	90	65	65	72	66	70.0	B-
7	410021007	Gary	117	110.5	65	69	43	67.15	C+
8	410021008	Helen	117	65	50	43	54	57.7	D
9	410021009	Igor	63	59.5	50	51	75	62.55	C
10	410021010	Jeff	117	110.5	117	53	75	80.35	A-
11	410021011	King	100	100	100	56	68	74.0	B
12	410021012	Leo	95	90	85	58	53	65.6	C
13	410021013	Mark	100	90	90	81	72	81.1	A-
14	410021014	Nancy	80	85	90	64	57	67.5	C+
15	410021015	Oliver	75	90	100	72	63	73.3	B
16	410021016	Peter	60	75	85	69	58	65.9	C
17	410021017	Quincy	85	80	90	44	39	54.3	D
18	410021018	Richar	90	85	80	52	48	60.3	D
19	410021019	Steve	100	90	90	83	61	77.3	B+
20	410021020	Tom	100	100	100	76	74	82.4	A-

R

```
● clara@clara-HP-Laptop-15:~/Downloads/HW2_PL_Track/PL_Track$ Rscript R.R
```

	NO	ID	Name	HW1	HW2	HW3	Midterm	Final	Score	Grade
1	1	410021001	Alan	90	84.5	117	60	66	73.55	B
2	2	410021002	Bob	85	49.0	80	57	64	64.10	C
3	3	410021003	Carrie	90	110.5	117	68	62	76.95	B+
4	4	410021004	David	117	85.0	0	44	55	55.40	D
5	5	410021005	Ethan	85	56.0	50	57	67	63.00	C
6	6	410021006	Frank	90	65.0	65	72	66	70.00	B-
7	7	410021007	Gary	117	110.5	65	69	43	67.15	C+
8	8	410021008	Helen	117	65.0	50	43	54	57.70	D
9	9	410021009	Igor	63	59.5	50	51	75	62.55	C
10	10	410021010	Jeff	117	110.5	117	53	75	80.35	A-
11	11	410021011	King	100	100.0	100	56	68	74.00	B
12	12	410021012	Leo	95	90.0	85	58	53	65.60	C
13	13	410021013	Mark	100	90.0	90	81	72	81.10	A-
14	14	410021014	Nancy	80	85.0	90	64	57	67.50	C+
15	15	410021015	Oliver	75	90.0	100	72	63	73.30	B
16	16	410021016	Peter	60	75.0	85	69	58	65.90	C
17	17	410021017	Quincy	85	80.0	90	44	39	54.30	D
18	18	410021018	Richar	90	85.0	80	52	48	60.30	C-
19	19	410021019	Steve	100	90.0	90	83	61	77.30	B+
20	20	410021020	Tom	100	100.0	100	76	74	82.40	A-

5. Discussion

For Python, R and Java, we didn't encounter many problems. It was easier to write compared to ML and Prolog. For ML, it took us a lot of time to figure out how to do "for" loops like other languages. ML is a language for functional programming language, it is more popular among compiler writers and programming language researchers. So using ML to operate data in CSV file isn't that easy. As for prolog, it's more powerful when associated with AI and computational linguistics. So using it to operate data in CSV is also inappropriate. But after lots of studies and discussion, we finally figured out how to write these two languages. But unfortunately, we are not able to read the CSV file and operate it. But we believed with more given time, we may find out some way to operate the CSV file.