Assignment 2: White Box Testing

SEG 3103 [Z] - Software Quality Assurance

Summer 2021
University of Ottawa

Course Coordinator: Professor Andrew Forward
Teaching Assistants: Zahra Kakavand
Nazanin Bayati Chaleshtari
Henry Chen

Document completed by:

Akram El-Gaouny, 300109692
Patrick Loranger, 300112374

Thursday, June 10th 2021

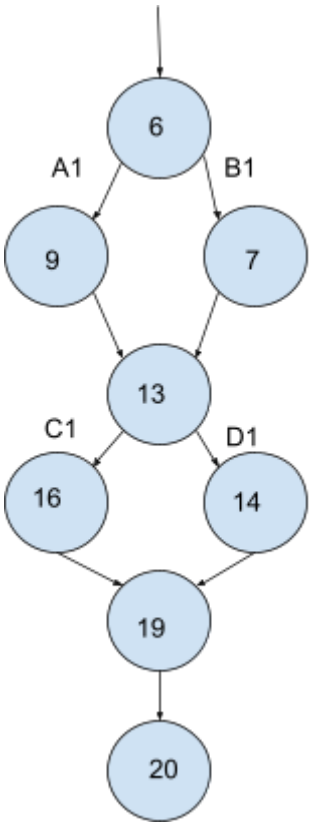
Problem 1:

Question 1.1: Draw the simplified control flow graph corresponding to each of the methods *percentage_grade*, *letter_grade*, and *numeric_grade*.

percentage_grade:

```
4   def percentage_grade(%{homework: homework, labs: labs, midterm: midterm, final: final}) do
5     avg_homework =
6       if Enum.count(homework) == 0 do
7         0
8       else
9         Enum.sum(homework) / Enum.count(homework)
10      end
11
12    avg_labs =
13      if Enum.count(labs) == 0 do
14        0
15      else
16        Enum.sum(labs) / Enum.count(labs)
17      end
18
19    mark = 0.2 * avg_labs + 0.3 * avg_homework + 0.2 * midterm + 0.3 * final
20    round(mark * 100)
21  end
```

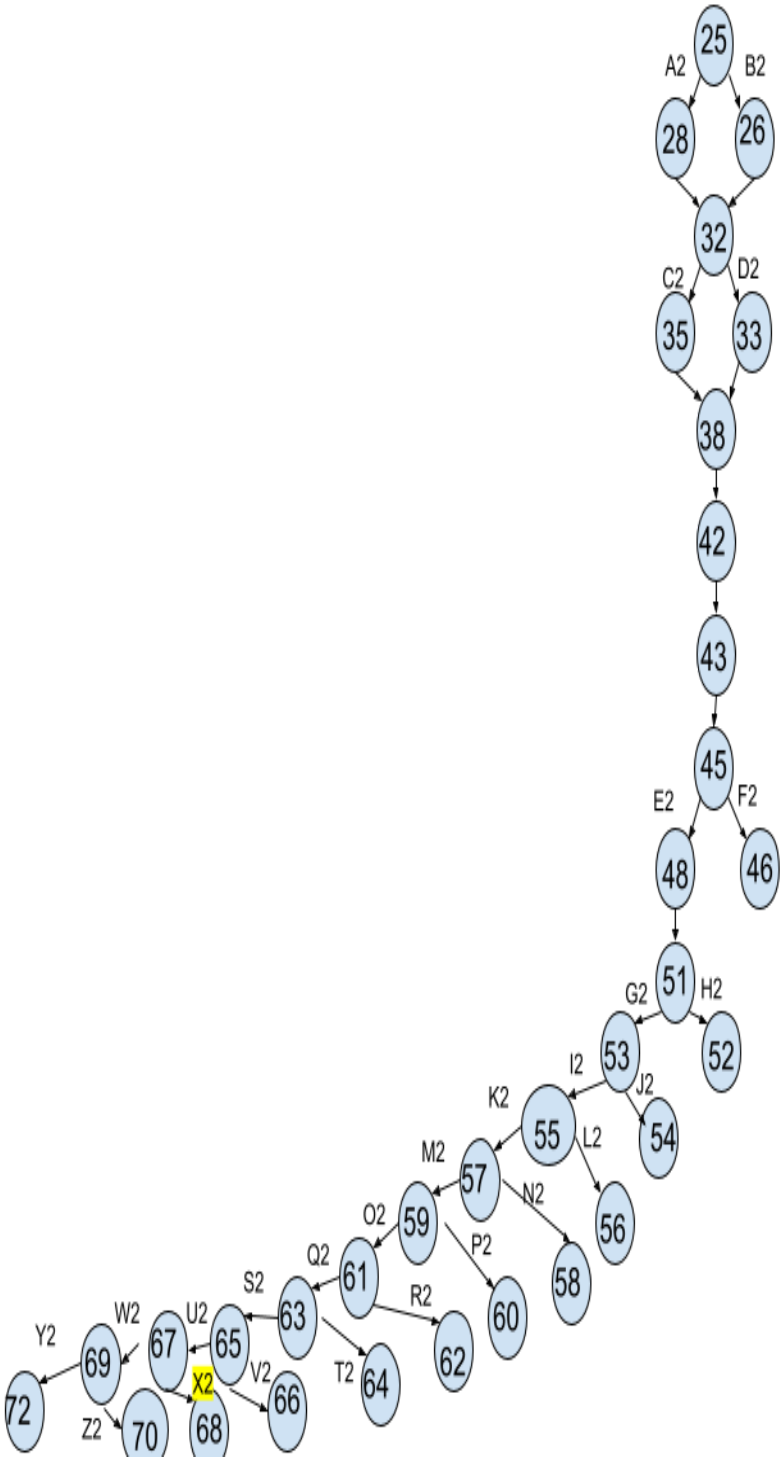
Please Note That The diagram below is based on the Line Number of the picture above.
At a condition Right mean True and Left Means False



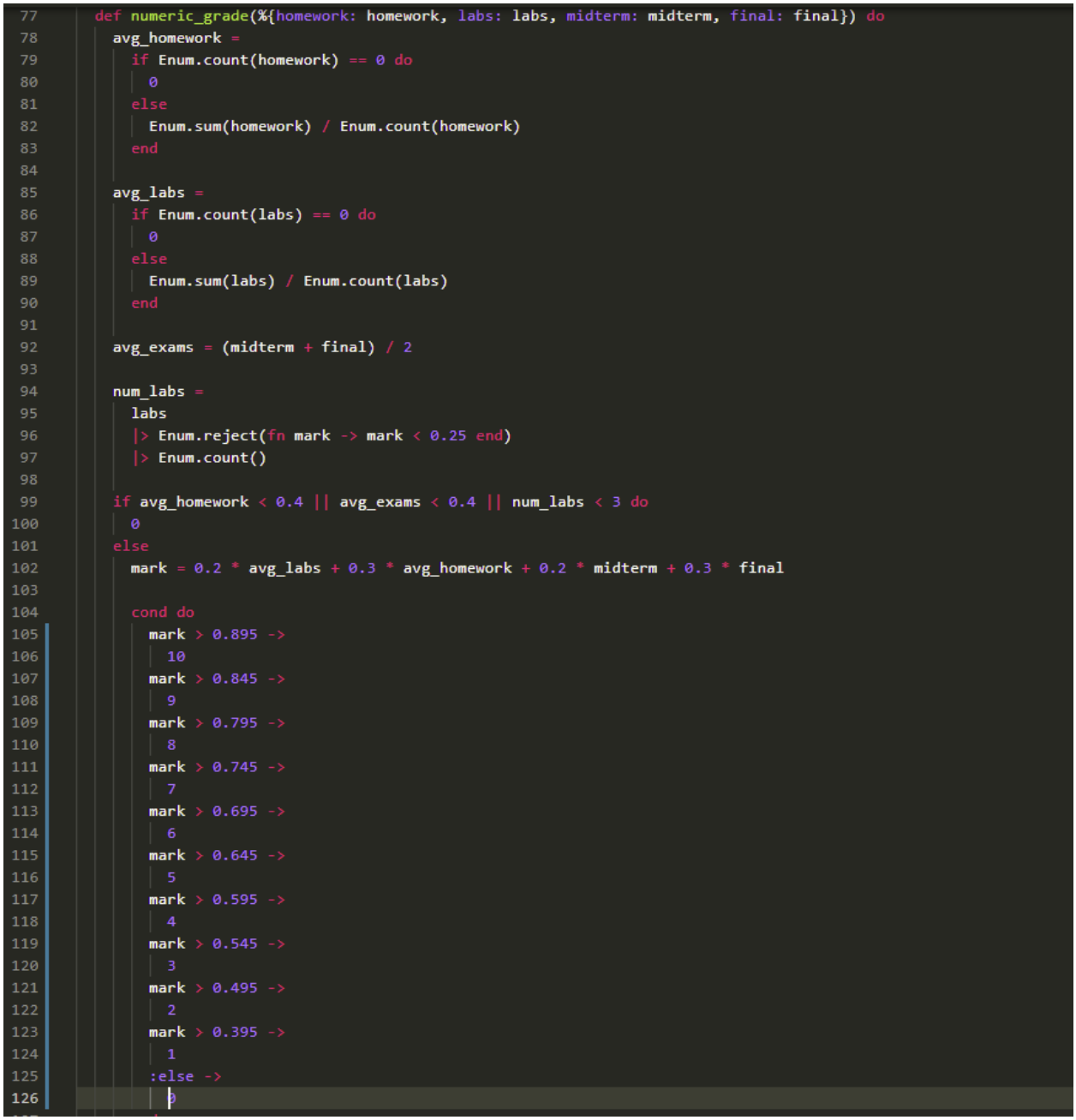
Letter_grade:

```elixir
def letter_grade(%{homework: homework, labs: labs, midterm: midterm, final: final}) do
  avg_homework =
    if Enum.count(homework) == 0 do
      0
    else
      Enum.sum(homework) / Enum.count(homework)
    end

  avg_labs =
    if Enum.count(labs) == 0 do
      0
    else
      Enum.sum(labs) / Enum.count(labs)
    end

  avg_exams = (midterm + final) / 2

  num_labs =
    labs
    |> Enum.reject(fn mark -> mark < 0.25 end)
    |> Enum.count()

  if avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3 do
    "EIN"
  else
    mark = 0.2 * avg_labs + 0.3 * avg_homework + 0.2 * midterm + 0.3 * final

    cond do
      mark > 0.895 ->
      "A+"
      mark > 0.845 ->
      "A"
      mark > 0.795 ->
      "A-"
      mark > 0.745 ->
      "B+"
      mark > 0.695 ->
        "B"
      mark > 0.645 ->
        "C+"
      mark > 0.595 ->
        "C"
      mark > 0.545 ->
        "D+"
      mark > 0.495 ->
        "D"
      mark > 0.395 ->
        "E"
      :else ->
        "F"
```

Please Note That The diagram below is based on the Line Number of the picture above.
At a condition Right mean True and Left Means False

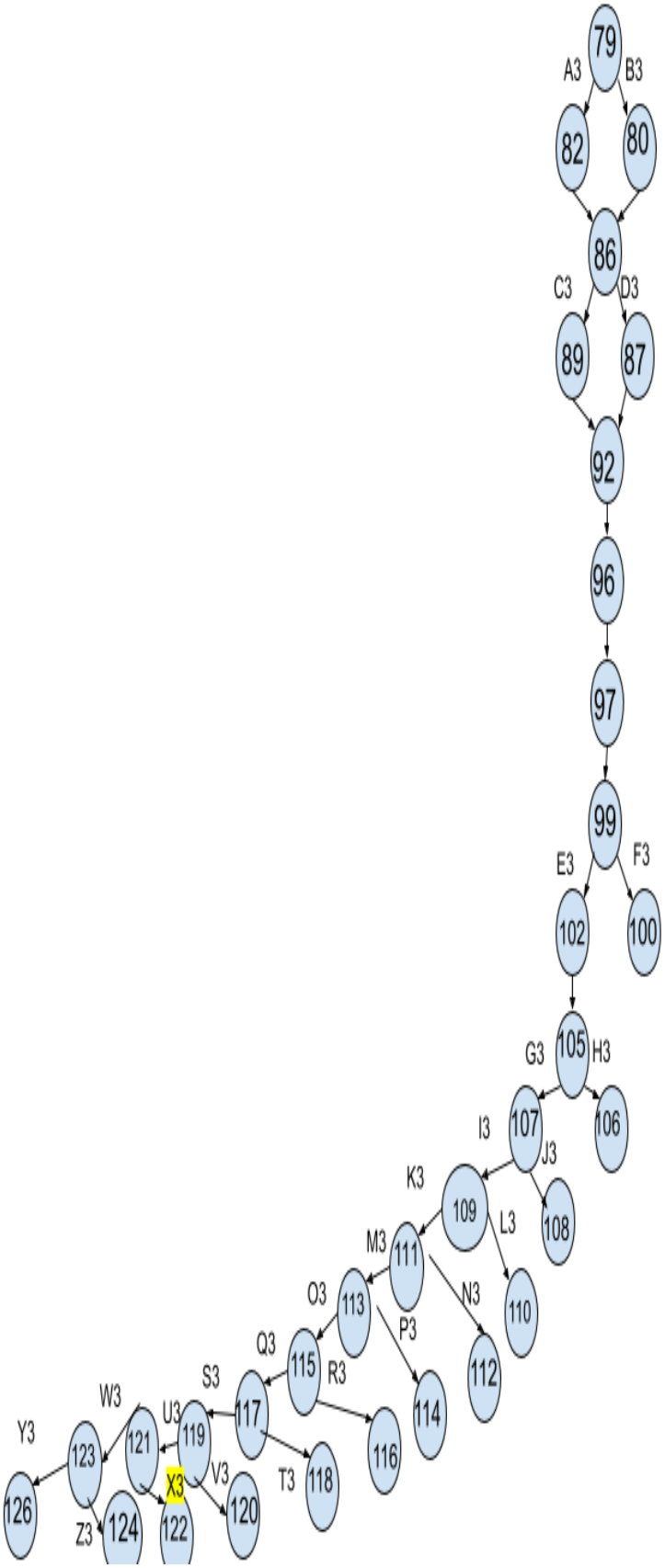

25
A2
B2
28
26
32
C2
D2
35
33
38
42
43
45
E2
F2
48
46
51
G2
H2
53
52
I2
J2
K2
55
54
L2
M2
57
56
N2
O2
59
58
P2
Q2
61
60
R2
S2
63
62
T2
64
U2
65
V2
66
W2
67
X2
68
Y2
69
Z2
70
72

numeric_grade:



```
77  def numeric_grade(%{homework: homework, labs: labs, midterm: midterm, final: final}) do
78    avg_homework =
79      if Enum.count(homework) == 0 do
80        0
81      else
82        Enum.sum(homework) / Enum.count(homework)
83      end
84
85    avg_labs =
86      if Enum.count(labs) == 0 do
87        0
88      else
89        Enum.sum(labs) / Enum.count(labs)
90      end
91
92    avg_exams = (midterm + final) / 2
93
94    num_labs =
95      labs
96      |> Enum.reject(fn mark -> mark < 0.25 end)
97      |> Enum.count()
98
99    if avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3 do
100     0
101   else
102     mark = 0.2 * avg_labs + 0.3 * avg_homework + 0.2 * midterm + 0.3 * final
103
104     cond do
105       mark > 0.895 ->
106         10
107       mark > 0.845 ->
108         9
109       mark > 0.795 ->
110         8
111       mark > 0.745 ->
112         7
113       mark > 0.695 ->
114         6
115       mark > 0.645 ->
116         5
117       mark > 0.595 ->
118         4
119       mark > 0.545 ->
120         3
121       mark > 0.495 ->
122         2
123       mark > 0.395 ->
124         1
125       :else ->
126         0
```

Please Note That The diagram below is based on the Line Number of the picture above.
At a condition Right mean True and Left Means False


79
A3
B3
82
80
86
C3
D3
89
87
92
96
97
99
E3
F3
102
100
105
G3
H3
107
106
I3
J3
K3
109
108
L3
M3
111
110
N3
O3
113
112
P3
Q3
115
114
R3
S3
117
116
W3
U3
119
118
T3
Y3
123
121
V3
120
X3
126
Z3
124
122

Question 1.2: Provide a white box test design for 100% branch coverage of the methods percentage_grade, letter_grade, and numeric_grade. Your test suite will be evaluated on the number of its test cases (try to have the smallest possible number of test cases that will allow 100% branch coverage). Use the following template for your test case design:

Notes For Reading the Table:

- The Conditional Branches are Labeled in the corresponding control-flow diagrams above (Ex: B1, D2, etc.)
- For the conditions, I put the node where there's a condition and what condition was evaluated. For example, (6-True) means that the condition in the if statement at line 6 of the code provided above was executed as true.
- Some conditions have multiple T/F, for example (45-True,True,True). What this means is that in line 45 there were three conditions and all of these conditions were true for a specific test

Test Case Number	Test Data	Expected Results	Conditions Covered	Branches Covered
percentage_grade Tests				
1	Homework: [] labs: [] midterm : 0.70 final: 0.70	35	6-True 13-True	B1, D1
2	Homework: [1,1] Labs: [1, 0.6,1,1] midterm : 0.70 final: 0.70	83	6-False 13- False	A1, C1
letter_grade Tests				
3	Homework: [] labs: [] midterm : 0.30 final: 0.30	"EIN"	25-True 32-True 45 - (True, True, True)	B2, D2, F2
4	Homework: [0.85,0.90,0.90] labs: [0.9, 0.9, 0.9] midterm : 1 final: 1	"A+"	25 - False 32 - False 45- (False, False, False) 51 - True	A2,C2,E2,H2
5	Homework: [0.85,0.86,0.82] labs: [0.9, 1, 0.83] midterm : 0.87	"A"	25-False 32-False 45-(False,False,Fals	A2, C2, E2,G2,J2

	final: 0.88		e) 51-False 53-True	
6	Homework: [1,1] Labs: [1, 0.6, 1,1] midterm : 0.70 final: 0.70	“A-”	25-False 32-False 45-(False,False,False) 51-False 53-False 55-True	A2, C2, E2,G2,I2,L2
7	Homework: [0.75,0.75] Labs: [0.7, 0.6, 0.75, 0.8] midterm : 0.75 final: 0.78	“B+”	25-False 32-False 45-(False,False,False) 51-False 53-False 55-False 57-True	A2, C2, E2,G2, I2, K2, N2
8	Homework: [0.7,0.74] Labs: [0.7, 0.6, 1,1] midterm : 0.73 final: 0.71	“B”	25-False 32-False 45-(False,False,False) 51-False 53-False 55-False 57-False 59-True	A2, C2, E2, G2, I2, K2, M2, P2
9	Homework: [0.8,0.50] Labs: [0.65, 0.6, 0.7,0.67] midterm : 0.68 final: 0.66	“C+”	25-False 32-False 45-(False,False,False) 51-False 53-False 55-False 57-False 59-False 61-True	A2, C2, E2, G2, I2, K2, M2, O2, R2
10	Homework: [0.6,0.5] Labs: [0.61, 0.6, 0.65,0.62] midterm : 0.63 final: 0.64	“C”	25-False 32-False 45-(False,False,False) 51-False 53-False 55-False 57-False	A2, C2, E2, G2, I2, K2, M2, O2, Q2, T2

			59-False 61-False 63 -True	
11	Homework: [0.56,0.53], labs: [0.56, 0.56, 0.57,0.49] midterm: 0.55 final: 0.545	“D+”	25-False 32-False 45-(False,False,False) 51-False 53-False 55-False 57-False 59-False 61-False 63 -False 65 -True	A2, C2, E2, G2, I2, K2, M2, O2, Q2, S2, V2
12	homework: [0.52,0.51], labs: [0.52, 0.49, 0.51,0.49], midterm: 0.51, final: 0.52	“D”	25-False 32-False 45-(False,False,False) 51-False 53-False 55-False 57-False 59-False 61-False 63 -False 65 - False 67 - True	A2, C2, E2, G2, I2, K2, M2, O2, Q2, S2, U2, X2
13	homework: [0.42,0.43], labs: [0.4, 0.42, 0.44,0.41], midterm: 0.42, final: 0.44	“E”	25-False 32-False 45-(False,False,False) 51-False 53-False 55-False 57-False 59-False 61-False 63 -False 65 - False 67 - False 69 - True	A2, C2, E2, G2, I2, K2, M2, O2, Q2, S2, U2, W2, Z2
14	homework: [0.42,0.43], labs: [0.25, 0.25, 0.25,0.25], midterm: 0.42, final: 0.44	“F”	25-False 32-False 45-(False,False,False)	A2, C2, E2, G2, I2, K2, M2, O2, Q2, S2, U2, W2,

			51-False 53-False 55-False 57-False 59-False 61-False 63 -False 65 False 67 -False 69 -False	Y2
Numeric_grade Tests				
15	Homework: [] labs: [] midterm : 0.30 final: 0.30	0	79 - True 86 -True 99 - (True, True, True)	B3, D3, F3
16	Homework: [0.85,0.90,0.90] labs: [0.9, 0.9, 0.9] midterm : 1 final: 1	10	79-False 86-False 99 - (False,False,False) 105 - True	A3,C3,E3,H3
17	Homework: [0.85,0.86,0.82] labs: [0.9, 1, 0.83] midterm : 0.87 final: 0.88	9	79-False 86-False 99 - (False,False,False) 105 - False 107- True	A3, C3, E3,G3,J3
18	Homework: [1,1] Labs: [1, 0.6, 1,1] midterm : 0.70 final: 0.70	8	79-False 86-False 99 - (False,False,False) 105 - False 107- False 109 -True	A3, C3, E3,G3,I3,L3
19	Homework: [0.75,0.75] Labs: [0.7, 0.6, 0.75, 0.8] midterm : 0.75 final: 0.78	7	79-False 86-False 99 - (False,False,False) 105 - False 107- False 109- False 111-True	A3, C3, E3,G3, I3, K3, N3
20	Homework: [0.7,0.74] Labs: [0.7, 0.6, 1,1] midterm : 0.73	6	79-False 86-False 99 -	A3, C3, E3, G3, I3, K3, M3, P3

	final: 0.71		(False,False,False) 105 - False 107- False 109- False 111-False 113-True	
21	Homework: [0.8,0.50] Labs: [0.65, 0.6, 0.7,0.67] midterm : 0.68 final: 0.66	5	79-False 86-False 99 - (False,False,False) 105 - False 107- False 109- False 111-False 113-False 115-True	A3, C3, E3, G3, I3, K3, M3, O3, R3
22	Homework: [0.6,0.5] Labs: [0.61, 0.6, 0.65,0.62] midterm : 0.63 final: 0.64	4	79-False 86-False 99 - (False,False,False) 105 - False 107- False 109- False 111-False 113-False 115-False 117-True	A3, C3, E3, G3, I3, K3, M3, O3, Q3, T3
23	Homework: [0.56,0.53], labs: [0.56, 0.56, 0.57,0.49] midterm: 0.55 final: 0.545	3	79-False 86-False 99 - (False,False,False) 105 - False 107- False 109- False 111-False 113-False 115-False 117-False 119-True	A3, C3, E3, G3, I3, K3, M3, O3, Q3, S3, V3
24	homework: [0.52,0.51], labs: [0.52, 0.49, 0.51,0.49], midterm: 0.51, final: 0.52	2	79-False 86-False 99 - (False,False,False) 105 - False 107- False 109- False	A3, C3, E3, G3, I3, K3, M3, O3, Q3, S3, U3, X3

			111-False 113-False 115-False 117-False 119-False 121-True	
25	homework: [0.42,0.43], labs: [0.4, 0.42, 0.44,0.41], midterm: 0.42, final: 0.44	1	79-False 86-False 99 - (False,False,False) 105 - False 107- False 109- False 111-False 113-False 115-False 117-False 119-False 121-False 123 -True	A3, C3, E3, G3, I3, K3, M3, O3, Q3, S3, U3, W3, Z3
26	homework: [0.42,0.43], labs: [0.25, 0.25, 0.25,0.25], midterm: 0.42, final: 0.44	0	79-False 86-False 99 - (False,False,False) 105 - False 107- False 109- False 111-False 113-False 115-False 117-False 119-False 121-False 123 -False	A3, C3, E3, G3, I3, K3, M3, O3, Q3, S3, U3, W3, Y3

Question 1.3: Provide an Implementation of your test suite using ExUnit

Implemented in **grades/test/grades/calculator_test.exs**

The following screenshot shows that all the tests have passed for the Grades.Calculator Module

```
C:\Users\jaoun\Desktop\Summer 2021\SEG 3103\SEG3103_Assignments\Assignment2\grades>mix test test/grades/calculator_test.exs
..........................

Finished in 0.06 seconds
26 tests, 0 failures

Randomized with seed 709000
```

Question 1.4: What is the degree of statement coverage obtained? If you weren't able to achieve 100% coverage explain why. Please be sure to attach screenshots of your coverage results. Elixir's coverage tool is primitive, as it only provides statement level accuracy. *mix test --cover* How might you address the limitations of a testing tool that only provides statement level coverage?

The degree of statement coverage that we obtained was 100% as we were able to cover all the statements in the three methods. We were able to achieve that since we were aiming to achieve a 100% percent branch coverage which implies a 100% statement coverage.

To address the limitation of a tool that gives you only statement coverage, you really need to analyze the relationship between the type of coverage you want to achieve and the statement coverage that was given by the testing tool. For example, we designed our tests to achieve a 100% branch coverage, so we would expect the result of the statement coverage to be 100%. If that is not the case, then we know that we didn't achieve a 100% branch coverage, so you essentially get hints from the statement coverage tool and you use it to relate to the type of coverage you got.

Screenshots for Coverages:
percentage_grade :

cover/Elixir.Grades.Calculator.html

```
1     defmodule Grades.Calculator do
2       def percentage_grade(%{homework: homework, labs: labs, midterm: midterm, final: final}) do
3   2     avg_homework =
4           if Enum.count(homework) == 0 do
5             0
6           else
7   1         Enum.sum(homework) / Enum.count(homework)
8           end
9
10  2     avg_labs =
11          if Enum.count(labs) == 0 do
12            0
13          else
14  1         Enum.sum(labs) / Enum.count(labs)
15          end
16
17  2     mark = 0.2 * avg_labs + 0.3 * avg_homework + 0.2 * midterm + 0.3 * final
18  2     round(mark * 100)
19      end
20
```

Letter_grade:

```
21        def letter_grade(%{homework: homework, labs: labs, midterm: midterm, final: final}) do
22  12      avg_homework =
23            if Enum.count(homework) == 0 do
24              0
25            else
26  11          Enum.sum(homework) / Enum.count(homework)
27            end
28
29  12      avg_labs =
30            if Enum.count(labs) == 0 do
31              0
32            else
33  11          Enum.sum(labs) / Enum.count(labs)
34            end
35
36  12      avg_exams = (midterm + final) / 2
37
38  12      num_labs =
39            labs
40  42        |> Enum.reject(fn mark -> mark < 0.25 end)
41            |> Enum.count()
42
43  12      if avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3 do
44            "EIN"
45          else
46  11        mark = 0.2 * avg_labs + 0.3 * avg_homework + 0.2 * midterm + 0.3 * final
47
48  11        cond do
49              mark > 0.895 ->
50                "A+"
51
52  10          mark > 0.845 ->
53                "A"
54
55  9           mark > 0.795 ->
56                "A-"
57
58  8           mark > 0.745 ->
59                "B+"
60
61  7           mark > 0.695 ->
62                "B"
63
64  6           mark > 0.645 ->
65                "C+"
66
67  5           mark > 0.595 ->
68                "C"
69
70  4           mark > 0.545 ->
71                "D+"
72
73  3           mark > 0.495 ->
74                "D"
75
76  2           mark > 0.395 ->
77                "E"
78
79  1           :else ->
80                "F"
81            end
82          end
83        end
```

numeric_grade:

```
85        def numeric_grade(%{homework: homework, labs: labs, midterm: midterm, final: final}) do
86   12     avg_homework =
87            if Enum.count(homework) == 0 do
88              0
89            else
90   11         Enum.sum(homework) / Enum.count(homework)
91            end
92
93   12     avg_labs =
94            if Enum.count(labs) == 0 do
95              0
96            else
97   11         Enum.sum(labs) / Enum.count(labs)
98            end
99
100  12     avg_exams = (midterm + final) / 2
101
102  12     num_labs =
103           labs
104  42       |> Enum.reject(fn mark -> mark < 0.25 end)
105           |> Enum.count()
106
107  12     if avg_homework < 0.4 || avg_exams < 0.4 || num_labs < 3 do
108           0
109         else
110  11       mark = 0.2 * avg_labs + 0.3 * avg_homework + 0.2 * midterm + 0.3 * final
111
112  11       cond do
113             mark > 0.895 ->
114               10
115
116  10         mark > 0.845 ->
117               9
118
119  9          mark > 0.795 ->
120               8
121
122  8          mark > 0.745 ->
123               7
124
125  7          mark > 0.695 ->
126               6
127
128  6          mark > 0.645 ->
129               5
130
131  5          mark > 0.595 ->
132               4
133
134  4          mark > 0.545 ->
135               3
136
137  3          mark > 0.495 ->
138               2
139
140  2          mark > 0.395 ->
141               1
142
143  1          :else ->
144               0
145           end
146       end
147     end
148   end
```