Name: Munim Mowsum ; Roll : 201116 ; MIT 22nd Batch , IIT-DU

Class : Grade Calculator

Multiple Condition Coverage

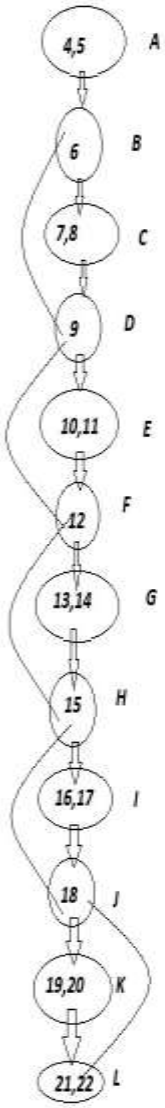
Method getGrade :

Test Case ID	Total>=80	Total>=70	Total>=60	Total>=50	Total<=50	Expected Result
1	True(90)	True	True	True	False	A+(failed)
2	False	True(78)	True	True	False	A(failed)
3	False	False	True(65)	True	False	B(failed)
4	False	False	False	True(51)	False	C(Pass)
5	False	False	False	False	True(49)	F(Pass)

Method getGPA :

Test Case ID	A+	A	B	C	else	Expected Result
1	True	False	False	False	False	4.00(failed)
2	False	True	False	False	False	3.50(failed)
3	False	False	True	False	False	3.00(failed)
4	False	False	False	True	False	2.00
5	False	False	False	False	True	0.0

Independent Path Testing



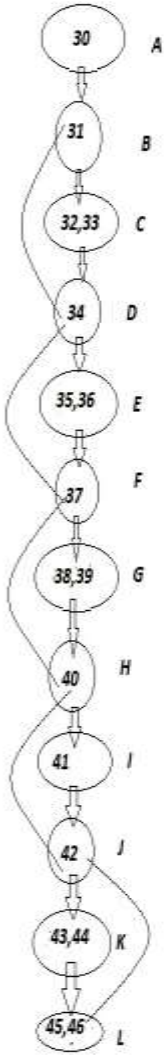
Method-getGrade:

V(G)=e-n+2*p=16-12+2*1=6

V(G)=Number of predicate nodes+1 = 5+1 = 6

(i) A,B,D,F,H,J,K,L
(ii) A,B,D,F,H,I,J,L
(iii) A,B,D,F,G,H,I,J,L
(iv) A,B,D,E,F,G,H,I,J,L
(v) A,B,C,D,E,F,G,H,I,J,L
(vi) A,B,D,F,H,I,J,K,L

Test Case ID	Input	Expected Result	Independentpath coveredby Testcase
1	49	F	A,B,D,F,H,J,K,L
2	55	C	A,B,D,F,H,I,J,L
3	65	B	A,B,D,F,G,H,I,J,L
4	75	A	A,B,D,E,F,G,H,I,J,L
5	85	A+	A,B,C,D,E,F,G,H,I,J,L
6	50	C/F	A,B,D,F,H,I,J,K,L



Method-GetGPA:

V(G)=e-n+2*p=15-12+2*1=5

V(G)=Number of predicate nodes+1 = 5+1 = 6

V(G)=Number of regions= 5 (R1=R5)

(i)A,B,D,F,H,J,K,L

(ii)A,B,D,F,H,I,J,L

(iii)A,B,D,F,G,H,I,J,L

(iv)A,B,D,E,F,G,H,I,J,L

(v)A,B,C,D,E,F,G,H,I,J,L

Test Case ID	Input	Expected Result	Independentpath coveredby Testcase
7	F	0.0	A,B,D,F,H,J,K,L
8	C	2.00	A,B,D,F,H,I,J,L
9	B	3.00	A,B,D,F,G,H,I,J,L
10	A	3.50	A,B,D,E,F,G,H,I,J,L
11	A+	4.00	A,B,C,D,E,F,G,H,I,J,L

Data Flow Testing:

Method-getGrade:

Variable -Total :

Pattern	Line Number	Explanation
~du	4	Normal case : Allowed
uu	6-8,9-11,12-14,15-17,18-20	Normal case : Allowed
K~	22	Normal case : Allowed

Variable -Grade:

Pattern	Line Number	Explanation
~d	5	Normal case : Allowed
uu	7-8,10-11,13-14,16-17,19-20	Normal case : Allowed
uk	21	Allowed
K~	22	Allowed

Method-getGPA:

Variable -gpa:

Pattern	Line Number	Explanation
~du	30	Normal case : Allowed
uu	32-33,35-36,38-39,43	Normal case : Allowed
uk	45	Allowed
K~	46	Allowed

Variable -Grade:

Pattern	Line Number	Explanation
kd	28	Allowed

uu	31-32,34-35,37-38,40-41	Normal case : Allowed
K~	21	Allowed

Mutation Testing:

(N.B : There are already failures / bug in the program ; So mutation testing is not necessary , although I did this for more elaborate testing.)

Method-getGrade:

Test Case ID	Total =	Grade =	Initial Program Result	Mutant Result
1	>85	X	A+	X
2	==78	A+	A	A+
3	<70	B+	B	B+
4	<60	E	C	E
5	==50	Z	F	Z

Method-getGPA:

Test Case ID	Total =	Grade =	Initial Program Result	Mutant Result
1	7.00	A-	4.00	7.00
2	4.00	A+	3.50	4.00
3	2.50	B-	3.00	2.50
4	1.00	C-	2.00	1.00

5	0.10	Y	0.0	0.10

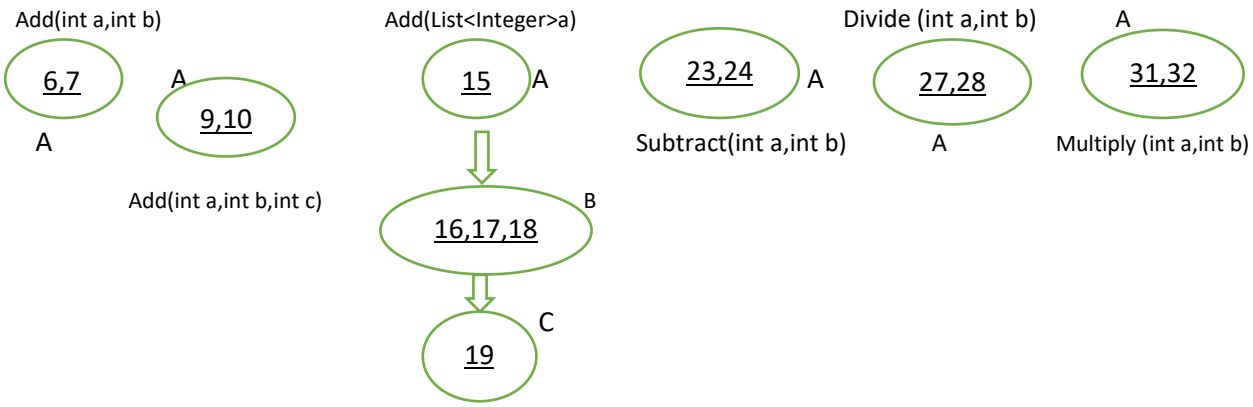
Class : Calculator

Multiple Condition Coverage

Lets take , a=2;b=2;c=2 .

Test Case ID	Method	Return	Expected Result
1	Add(int a,int b)	True	4
2	Add(int a,int b,int c)	True	6
3	Add(List<Integer>a)	True	6
4	Subtract(int a,int b)	True	0
5	Divide (int a,int b)	True	1
6	Multiply (int a,int b)	True	4

Independent Path Testing



Method	TestCase ID	a	b	c	Expected Result	Path
Add(int a,int b)	1	1	2	×	3	A
Add(int a,int b,int c)	2	1	2	3	6	A
Add(List<Integer>a)	3	List{1,2,3}			6	A,B,C
Subtract(int a,int b)	4	1	2	×	-1	A
Divide (int a,int b)	5	1	2	×	0.5	A
Multiply (int a,int b)	6	1	2	×	2	A

Data Flow Testing:

Method	Test Case ID	Variable	Pattern	Line Number	Explanation
Add(int a,int b)	1	int a,int b	~d uu kk	5 6 7	Allowed “ “
Add(int a,int b,int c)	2	int a,int b,int c	Kd uu kk	9 10 11	Allowed “ “
Add(List<Integer>a)	3	List<Integer>a	~d uu kk	13 16 21	Allowed “ “
		Integer temp	~du uu kk	15 17,19 21	Allowed “ “
Subtract(int a,int b)	4	int a,int b	Kd uu kk	22 23 24	Allowed “ “
Divide (int a,int b)	5	int a,int b	Kd uu kk	26 27 28	Allowed “ “

Multiply (int a,int b)	6	int a,int b	Kd uu kk	30 31 32	Allowed “ “

Mutation Testing:

Let’s take a=1;b=2;c=3

Method	TestCase ID	Alteration	InitialProgram Result	Mutant Result
Add(int a,int b)	1	return a-b	3	-1
Add(int a,int b,int c)	2	return a-b-c	6	-4
Add(List<Integer>a)	3	Temp -=num	6	-6
Subtract(int a,int b)	4	return a+b	-1	3
Divide (int a,int b)	5	return a*b	0.5	2
Multiply (int a,int b)	6	return a/b	2	0.5