**A close up of a logo

Description automatically generated**

**SOFTWARE TESTING**

**Assignment 2**

**Section 2**

**Date: 5-15-2020**

Sameen Zahra BSE173137

Mishal khan BSE173032

Noor Huda BSE173120

**Submitted To**

**Sir Sameer Obaid**

Contents

[**1** **Case Study:** 3](#_Toc40507351)

[**1.1** **Introduction** 3](#_Toc40507352)

[**1.2** **Brief Introduction** 3](#_Toc40507353)

[**2** **Identified Test Cases:** 4](#_Toc40507354)

[**2.1** **Test case 01** 4](#_Toc40507355)

[**2.2** **Test case 02** 4](#_Toc40507356)

[**2.3** **Test case 03** 4](#_Toc40507357)

[**3** **Black Box Testing:** 4](#_Toc40507358)

[**3.1** **Worst case BVA** 4](#_Toc40507359)

[**3.1.1** **Function 01** 4](#_Toc40507360)

[**3.1.2** **Function 02** 7](#_Toc40507361)

[**3.1.3** **Function 03** 7](#_Toc40507362)

[**4** **Strong robust equivalence classes:** 7](#_Toc40507363)

[**4.1** **Function 01 ((static public String marks (int ass, int mid, int finals))** 7](#_Toc40507364)

[**5** **Strong robust equivalence classes AND Robust worst case BVA :** 8](#_Toc40507365)

[**6** **Observation:** 8](#_Toc40507366)

# **Case Study:**

## **Introduction**

Accept three integers which are supposed to be the three evaluation criteria and determine if the three values represent grade A,B,C,D,F Derive test cases for your program based on WORSTBVA approach, and strong robust equivalence class. execute the test cases and discuss the results.

## **Brief Description**

In this project we are calculating the GPA by giving 3 different evaluation criteria’s that are midterm marks, assignment marks and finals marks also we have assigned boundaries to marks. For assignments 1-20, for midterms 1-20 and for finals 1-60and assigning them grades. In our project we have 3 functions 1st function marks with 3 parameters. 2nd function grades with 1 parameter. 3rdGPAwith 1 parameter. In marks function 3 parameters midterm marks, assignment marks, and final marks is taking input from test cases and calculating sum of assignment marks, midterm marks and finals marks then saving the sum of three inputs in some variable and then we are passing value of sum in 2nd function which is grade now in our 2nd function with one parameter taking value of sum and assigning them grade by following conditions if sum grater or equals to 90 assign grade A, if sum grater or equals 80 assign grade B, if sum grater or equals to 70 assign grade C, if sum grater or equals to 50 assign grade D, if sum is below 50 then assign grade F. These grades are passing in our 3rd function that is GPA. In 3rd function we assign GPA according to grade.

If grade is equal to A assign GPA equals 4.01, If grade is equal to B assign GPA equals 3.33, If grade is equal to C assign GPA equals 2.33, If grade is equal to D assign GPA equals 1.2 ,grade equals F assign F.

With above description we have generated our test cases with worst BVA and strong robust equivalence class. Mentioned below

# **Identified Test Cases:**

## **Test case 01**

**static public String marks (int ass, int mid, int finals){code on GitHub}**

In this function it will take 3 parameters that is ass, mid, finals which will be taken from test cases (ass, mid, finals)and calculate the sum of these 3 parameters that will help in assigning the grade to student courses .After calculating sum program will save this value in var and pass this value to 2nd function and grade will returned from 2nd function and passed to 3rd function.

## **Test case 02**

**static public String grade (int result ) ){code on GitHub}**

In this function it will take 1 parameters that is result , taking value from main function and assigning the grades on sum according to defined criteria that is if sum grater or equals to 90 assign grade A, if sum grater or equals 80 assign grade B, if sum grater or equals to 70 assign grade C, if sum grater or equals to 50 assign grade D, if sum is below 50 then assign grade F.

## **Test case 03**

**static public String gpa(String a ) { code on GitHub}**

In this function it will take 1 parameters that is some string a , taking grade from main function and assigning the gpa on Grade according to defined criteria that If grade is equal to A assign GPA equals 4.01, If grade is equal to B assign GPA equals 3.33, If grade is equal to C assign GPA equals 2.33, If grade is equal to D assign GPA equals 1.2 ,grade equals F assign F. And then it returns value to function.

# **Black Box Testing:**

## **Worst case BVA**

### **Function 01**

**Total test cases**= 5n = 53=125   
**Half test cases implemented** = 66 ,3 more then half are tested   
**Input Values:**  
**ass:** min = 1, min+ = 2, normal= 13, max- = 19, max = 20  
**mid:**min = 1, min+ = 2, normal = 13, max- = 19, max = 20

**finals:** min = 1, min+ = 2, normal = 35, max- = 59, max = 6

|  |  |
| --- | --- |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case | Input 1 | Input 2 | Input 3 | | Output |
| 1 | 1 | 1 | | 1 | F |
| 2 | 1 | 1 | | 2 | F |
| 3 | 1 | 1 | | 35 | F |
| 4 | 1 | 1 | | 59 | 1.2 |
| 5 | 1 | 1 | | 60 | 1.2 |
| 6 | 1 | 2 | | 1 | F |
| 7 | 1 | 2 | | 2 | F |
| 8 | 1 | 2 | | 35 | F |
| 9 | 1 | 2 | | 59 | 1.2 |
| 10 | 1 | 2 | | 60 | 1.2 |
| 11 | 1 | 13 | | 1 | F |
| 12 | 1 | 13 | | 2 | F |
| 13 | 1 | 13 | | 35 | F |
| 14 | 1 | 13 | | 59 | 2.33 |
| 15 | 1 | 13 | | 60 | 2.33 |
| 16 | 1 | 19 | | 1 | F |
| 17 | 1 | 19 | | 2 | F |
| 18 | 1 | 19 | | 35 | 1.2 |
| 19 | 1 | 19 | | 59 | 2.33 |
| 20 | 1 | 19 | | 60 | 3.33 |
| 21 | 1 | 20 | | 1 | F |
| 22 | 1 | 20 | | 2 | F |
| 23 | 1 | 20 | | 35 | 1.2 |
| 24 | 1 | 20 | | 59 | 3.33 |
| 25 | 1 | 20 | | 60 | 3.33 |
| 26 | 2 | 1 | | 1 | F |
| 27 | 2 | 1 | | 2 | F |
| 28 | 2 | 1 | | 35 | F |
| 29 | 2 | 1 | | 59 | 1.2 |
| 30 | 2 | 1 | | 60 | 1.2 |
| 31 | 2 | 2 | | 1 | F |
| 32 | 2 | 2 | | 2 | F |
| 33 | 2 | 2 | | 35 | F |
| 34 | 2 | 2 | | 59 | 1.2 |
| 35 | 2 | 2 | | 60 | 1.2 |
| 36 | 2 | 13 | | 1 | F |
| 37 | 2 | 13 | | 2 | F |
| 38 | 2 | 13 | | 35 | 1.2 |
| 39 | 2 | 13 | | 59 | 2.33 |
| 40 | 2 | 13 | | 60 | 2.33 |
| 41 | 2 | 19 | | 1 | F |
| 42 | 2 | 19 | | 2 | F |
| 43 | 2 | 19 | | 35 | 1.2 |
| 44 | 2 | 19 | | 59 | 3.33 |
| 45 | 2 | 19 | | 60 | 3.33 |
| 46 | 2 | 20 | | 1 | F |
| 47 | 2 | 20 | | 2 | F |
| 48 | 2 | 20 | | 35 | 1.2 |
| 49 | 2 | 20 | | 59 | 3.33 |
| 50 | 2 | 20 | | 60 | 3.33 |
| 51 | 13 | 1 | | 1 | F |
| 52 | 13 | 1 | | 2 | F |
| 53 | 13 | 1 | | 35 | F |
| 54 | 13 | 1 | | 59 | 2.33 |
| 55 | 13 | 1 | | 60 | 2.33 |
| 56 | 13 | 2 | | 1 | F |
| 57 | 13 | 2 | | 2 | F |
| 58 | 13 | 2 | | 35 | 1.2 |
| 59 | 13 | 2 | | 59 | 2.33 |
| 60 | 13 | 2 | | 60 | 2.33 |
| 61 | 13 | 13 | | 1 | F |
| 62 | 13 | 13 | | 2 | F |
| 63 | 13 | 13 | | 35 | 1.2 |
| 64 | 13 | 13 | | 59 | 3.33 |
| 65 | 13 | 13 | | 60 | 3.33 |
| 66 | 13 | 19 | | 1 | F |
| 67 | 13 | 19 | | 2 | F |
| 68 | 13 | 19 | | 35 | 1.2 |
| 69 | 13 | 19 | | 59 | 4.01 |
| 70 | 13 | 19 | | 60 | 4.01 |
| 71 | 13 | 20 | | 1 | F |
| 72 | 13 | 20 | | 2 | F |
| 73 | 13 | 20 | | 35 | 1.2 |
| 74 | 13 | 20 | | 59 | 4.01 |
| 75 | 13 | 20 | | 60 | 4.01 |
| 76 | 19 | 1 | | 1 | F |
| 77 | 19 | 1 | | 2 | F |
| 78 | 19 | 1 | | 35 | 1.2 |
| 79 | 19 | 1 | | 59 | 2.33 |
| 80 | 19 | 1 | | 60 | 3.33 |
| 81 | 19 | 2 | | 1 | F |
| 82 | 19 | 2 | | 2 | F |
| 83 | 19 | 2 | | 35 | 1.2 |
| 84 | 19 | 2 | | 59 | 3.33 |
| 85 | 19 | 2 | | 60 | 3.33 |
| 86 | 19 | 13 | | 1 | F |
| 87 | 19 | 13 | | 2 | F |
| 88 | 19 | 13 | | 35 | 1.2 |
| 89 | 19 | 13 | | 59 | 4.01 |
| 90 | 19 | 13 | | 60 | 4.01 |
| 91 | 19 | 19 | | 1 | F |
| 92 | 19 | 19 | | 2 | F |
| 93 | 19 | 19 | | 35 | 2.33 |
| 94 | 19 | 19 | | 59 | 4.01 |
| 95 | 19 | 19 | | 60 | 4.01 |
| 96 | 19 | 20 | | 1 | F |
| 97 | 19 | 20 | | 2 | F |
| 98 | 19 | 20 | | 35 | 2.33 |
| 99 | 19 | 20 | | 59 | 4.01 |
| 100 | 19 | 20 | | 60 | 4.01 |
| 101 | 20 | 1 | | 1 | F |
| 102 | 20 | 1 | | 2 | F |
| 103 | 20 | 1 | | 35 | 1.2 |
| 104 | 20 | 1 | | 59 | 3.33 |
| 105 | 20 | 1 | | 60 | 3.33 |
| 106 | 20 | 2 | | 1 | F |
| 107 | 20 | 2 | | 2 | F |
| 108 | 20 | 2 | | 35 | 1.2 |
| 109 | 20 | 2 | | 59 | 3.33 |
| 110 | 20 | 2 | | 60 | 3.33 |
| 111 | 20 | 13 | | 1 | F |
| 112 | 20 | 13 | | 2 | F |
| 113 | 20 | 13 | | 35 | 1.2 |
| 114 | 20 | 13 | | 59 | 4.01 |
| 115 | 20 | 13 | | 60 | 4.01 |
| 116 | 20 | 19 | | 1 | F |
| 117 | 20 | 19 | | 2 | F |
| 118 | 20 | 19 | | 35 | 2.33 |
| 119 | 20 | 19 | | 59 | 4.01 |
| 120 | 20 | 19 | | 60 | 4.01 |
| 121 | 20 | 20 | | 1 | F |
| 122 | 20 | 20 | | 2 | F |
| 123 | 20 | 20 | | 35 | 2.3 |
| 124 | 20 | 20 | | 59 | 4.01 |
| 125 | 20 | 20 | | 60 | 4.01 |

### **Function 02**

**Total test cases**= 5n = 51=5

### **Function 03**

**Total test cases**= 5n = 51=5

# **Strong robust equivalence classes:**

### **Function 01 ((static public String marks (int ass, int mid, int finals))**

**Total test cases** = 4\*4\*5=80

**Test data** = Enter 3 inputs values of type integer.  
**Class 1(ass** -1 to 21)**= -1-5,5-11,11-17,17-21**

**Class 2(mid** -1 to 21)**= -1-5,5-11,11-17,17-21**

**Class 3(finals** -1 to 61) **= -1-19,19-29,29-39,39-49,49-61**

# **Strong robust equivalence classes AND Robust worst case BVA :**

**Robust worst case BVA**

**Total test cases** = 7n = 73=343

**Strong robust equivalence classes**

**Total test cases** = 4\*4\*5=80

# **Observation:**

In above understanding Robust worst case BVA give almost thrice test cases then Strong robust equivalence classes with reputation of cases. And Strong robust equivalence classes give less test cases after considering all boundaries and beyond boundaries values so hare test cases are reduced in comparison with Robust worst case BVA