Ain Shams University
Faculty of Engineering
Computer & System Engineering Program

Software Testing
CSE 337s

Project Phase 1

Team :8

Mohamed Abdel Hamid Abdel Hares	*2001087*
Anas Hamed Ahmed Hamed	*2001584*
Kareem khaled abdelazim mohamed	*2001091*
Mariam ahmed abd elgalil	*2001273*
Sara Ashraf Abdelhakam	*2000337*
Ahmed gamal helmy	*2000082*

GitHub Link: *https://github.com/kareemkhaledd/Student-grading-system.git*

Testing Type Details: Test Parse File Module
Tester Name: Mohamed Abdel Hamid
Test Environment Details

Scen#	Scenario Description	Req #	Cond #	Test Data **Each line is considered input line**	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/ Fail (Y/N)
1	Test throwing exception when enter not completed attributes For Subject	11		**Input=** Testing,CSE321 Mohamed,2001087,5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *getSubject_notCompletedAttributesForStudent* method	Exception is thrown and message 'Invalid Subject Attributes' is printed.	Parsing of subject is not done and no student object is created and exception is thrown.	Error and message 'Invalid Subject Attributes' is printed.	Y
2	Test throwing exception when enter input with no subject	11		**Input=** Mohamed,2001087,5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *getSubject_noSubjectOrStudentsInputFile* method	Exception is thrown and message 'Invalid input file: It has no students or has no subject' is printed.	Parsing of subject is not done and no student object is created and exception is thrown.	Error and message 'Invalid input file: It has no students or has no subject' is printed.	Y
3	Test throwing exception when enter invalid subject name	1		**Input=** Test%%ing,CSE321,100 Mohamed,2001087,5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *getSubject_invalidName*	Exception is thrown and message 'Test%%ing is Invalid' is printed.	Parsing of subject is not done and no student object is created and exception is thrown.	Error and message 'Test%%ing is Invalid' is printed.	Y
4	Test throwing exception when enter invalid	2		**Input=** Testing,CSE&321,100 Mohamed,2001087,5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the	Exception is thrown and message 'CSE&321 is Invalid'	Parsing of subject is not done and no student object is	Error and message 'CSE&321 is Invalid' is printed.	Y

Testing Type Details: Test Parse File Module
Tester Name: Mohamed Abdel Hamid
Test Environment Details

Scen#	Scenario Description	Req #	Cond #	Test Data **Each line is considered input line**	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/ Fail (Y/N)
	subject code				expected and the actual results 4.run a junit test of *getSubject_invalidCode*	is printed.	created and exception is thrown.		
5	Test throwing exception when enter invalid max mark	3		**Input=** Testing,CSE321,-2 Mohamed,2001087,5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *getSubject_invalidMaxMark*	Exception is thrown and message ‘-2 is Invalid’ is printed.	Parsing of subject is not done and no student object is created and exception is thrown.	Error and message ‘-2 is Invalid’ is printed.	Y
6	Test throwing exception when enter invalid *Name and Code*	1,2		**Input=** Test%%ing,CSE3&&21,100 Mohamed,2001087,5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *getSubject_invalidNameAndCode*	Exception is thrown and message ‘Test%%ing is Invalid’ is printed.	Parsing of subject is not done and no student object is created and exception is thrown.	Error and message ‘Test%%ing is Invalid’ is printed.	Y
7	Test throwing exception when enter invalid *code and max*	2,3		**Input=** Testing,CS&&E321,-5 Mohamed,2001087,5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *getSubject_invalidCodeAndMax*	Exception is thrown and message ‘CS&&E321 is Invalid’ is printed.	Parsing of subject is not done and no student object is created and exception is thrown.	Error and message ‘CS&&E321 is Invalid’ is printed.	Y
8	Test parsing and storing of subject	1,2,3, 10,11		**Input=** Testing,CSE321,100	1.Create test method 2.Create test data	Object of subject is created which has	Parsing of subject data is done and it’s	Parsing is done: Object of subject is	Y

Testing Type Details: Test Parse File Module
Tester Name: Mohamed Abdel Hamid
Test Environment Details

Scen#	Scenario Description	Req #	Cond #	Test Data **Each line is considered input line**	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/ Fail (Y/N)
	when valid data is entered			Mohamed,2001087,5,5,5,5	3.use assertion to compare the expected and the actual results 4.run a junit test of *getSubject_storingWhenValidData*	name = 'Testing' and code = 'CSE321' and max mark = 100	ready for fill students.	created which has name = 'Testing' and code = 'CSE321' and max mark = 100	
9	Test parsing and storing of subject when invalid data is entered	1,2,3, 10,11		**Input=** Tes&&ting,CSE321,100 Mohamed,2001087,5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *getSubject_storingWhenInvalidData*	Object of subject is not created	Parsing of subject data is not done and exception is thrown.	Parsing is not done, and no object of subject is created.	Y
10	Test throwing exception when enter invalid student name	4		**Input=** Testing,CSE321,100 Moham&&ed,20001087,5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *fillStudents_invalidName*	Exception is thrown and message 'Moham&&ed is Invalid' is printed.	Parsing of students data is not done and it is not filled in subject object and exception is thrown.	Error and message 'Moham&&ed is Invalid' is printed.	Y
11	Test throwing exception when enter invalid student code	5		**Input=** Testing,CSE321,100 Mohamed,2001087,5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *fillStudent_invalidCode*	Exception is thrown and message '2001087 is Invalid' is printed.	Parsing of students data is not done and it is not filled in subject object and exception is thrown.	Error and message '2001087 is Invalid' is printed.	Y
12	Test throwing exception when	6,7,8, 9		**Input=** Testing,CSE321,100	1.Create test method 2.Create test data	Exception is thrown and message	Parsing of students data is not done and	Error and message '-5 is Invalid'	Y

Testing Type Details: Test Parse File Module
Tester Name: Mohamed Abdel Hamid
Test Environment Details

Scen#	Scenario Description	Req #	Cond #	Test Data **Each line is considered input line**	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/ Fail (Y/N)
	enter invalid student degrees			Mohamed,2001087,-5,5,5,5	3.use assertion to compare the expected and the actual results 4.run a junit test of *fillStudents_invalidDegrees*	‘-5 is Invalid’ is printed.	it is not filled in subject object and exception is thrown.	is printed.	
13	Test throwing exception when enter invalid student name and code	4,6,7, 8,9		**<u>Input</u>**= Testing,CSE321,100 Mohame&d,200&01087,-5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *fillStudents_invalidNameAndCode*	Exception is thrown and message ‘Mohame&dis Invalid’ is printed.	Parsing of students data is not done and it is not filled in subject object and exception is thrown.	Error and message ‘Mohame&d is Invalid’ is printed.	Y
14	Test throwing exception when enter invalid student code and degree	4,6,7, 8,9		**<u>Input</u>**= Testing,CSE321,100 Mohame&d,200&01087,-5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *fillStudents_invalidCodeAndDegree*	Exception is thrown and message ‘2001087 is Invalid’ is printed.	Parsing of students data is not done and it is not filled in subject object and exception is thrown.	Error and message ‘2001087 is Invalid’ is printed.	Y
15	Test parsing and storing of students when valid students data is entered	4,5,6, 7,8,9 10,11		**<u>Input</u>**= Testing,CSE321,100 Mohamed,20001087,5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *fillStudents_storingWhenValidData*	Object of subject is filled with student 1 which has name = ‘ Mohamed’ and code = ‘20001087’ and activities mark = 5 Practical mark = 5 Midterm mark = 5 Final mark = 5	Parsing of students data is done and filled in subject object and subject object is ready for calculating GPA.	Parsing is done: Object of subject is filled with student 1 which has name = Mohamed’ and code = ‘20001087’ and activities mark = 5 Practical mark = 5 Midterm mark = 5	Y

Testing Type Details: Test Parse File Module

Tester Name: Mohamed Abdel Hamid

Test Environment Details

Scen#	Scenario Description	Req #	Cond #	Test Data **Each line is considered input line**	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/ Fail (Y/N)
								Final mark = 5	
16	Test parsing and storing of students when valid students data is entered	4,5,6, 7,8,9 10,11		**Input=** Testing,CSE321,100 Moha&&med,2001087,5,5,5,5	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *fillStudent_storingWhenInvalidData*	Object of subject is not filled with objects	Parsing of students data is not done and not filled in subject object.	Parsing is not done, and object of subject is not filled with objects.	Y

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
1	Test correct Student number	5	1,2	**<u>Student number:</u>** 2001091x	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testValidNumberOf7NumbersAndCharacter_1* method	*CheckStudentNumber* method Return true	The student number can be saved in the prase module in its data field	*CheckStudentNumber* method Return true As expected	y
2	Test correct Student number	5	1	**<u>Student number:</u>** 20010911	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testValidNumberOf8Numbers* method	*CheckStudentNumber* method Return true	The student number can be saved in the prase module in its data field	*CheckStudentNumber* method Return true As expected	y

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
3	Test incorrect Student number	5	1	**<u>Student number:</u>** 2001091	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInvalidNumberLength* method	*CheckStudentNumber* method Return false	The student number can't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckStudentNumber* method Return false As expected	y
4	Test incorrect Student number	5	1	**<u>Student number:</u>** 2x010911	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInvalidNumberFormat_1* method	*CheckStudentNumber* method Return false	The student number can't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckStudentNumber* method Return false As expected	y

Testing Type Details: DataCheckModule

Tester Name:Kareem Khaled

Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
5	Test incorrect Student number	5	1	**Student number:** 20 10911	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInvalidNumberFormat_2* method	*CheckStudentNumber* method Return false	The student number can't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckStudentNumber* method Return false As expected	y
6	Test incorrect Student number	5	1	**Student number:** 200#0911	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInvalidNumberFormat_3* method	*CheckStudentNumber* method Return false	The student number can't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckStudentNumber* method Return false As expected	y

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
7	Test incorrect Student number	5	1	**Student number:** 0010911	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInvalidNumberFormat_4* method	*CheckStudentNumber* method Return false	The student number can't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckStudentNumber* method Return false As expected	y
8	Test incorrect Student number	5	1	**Student number:** 2001091#	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInvalidNumberFormat_6* method	*CheckStudentNumber* method Return false	The student number can't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckStudentNumber* method Return false As expected	y

Testing Type Details: DataCheckModule

Tester Name:Kareem Khaled

Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
9	Test correct ActivitiesMark	6	1	**mark**= 10 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testValidActivitiesMark_1* method	*CheckActivitiesMark* method Return true	The student activities mark can be saved in the prase module in its data field	*CheckActivitiesMark* method Return true As expected	y
10	Test correct ActivitiesMark	6	1	**mark**= 0 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testValidActivitiesMark_2* method	*CheckActivitiesMark* method Return true	The student activities mark can be saved in the prase module in its data field	*CheckActivitiesMark* method Return true As expected	y

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
11	Test correct ActivitiesMark	6	1	**mark=** 12 **fullMark**= 120	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testValidActivitiesMark_3* method	*CheckActivitiesMark* method Return true	The student activities mark can be saved in the prase module in its data field	*CheckActivitiesMark* method Return true As expected	y
12	Test incorrect ActivitiesMark	6	1	**mark=** -1 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInValidActivitiesMark_1* method	*CheckActivitiesMark* method Return false	The student activities mark can't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckActivitiesMark* method Return false As expected	y

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
13	Test incorrect ActivitiesMark	6	1	**<u>mark</u>**= 11 **<u>fullMark</u>**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInValidActivitiesMark_2* method	*CheckActivitiesMark* method Return false	The student activities mark can't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckActivitiesMark* method Return false As expected	y
14	Test incorrect ActivitiesMark	6	1	**<u>mark</u>**= 13 **<u>fullMark</u>**= 120	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInValidActivitiesMark_3* method	*CheckActivitiesMark* method Return false	The student activities mark can't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckActivitiesMark* method Return false As expected	y

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
15	Test correct OralPracticalMark	7	1	**mark=** 10 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testValidOralPracticalMark_1* method	*CheckOralPracticalMark* method Return true	The student *Oral Practical Mark* can be saved in the prase module in its data field	*CheckOralPracticalMark* method Return true As expected	y
16	Test correct OralPracticalMark	7	1	**mark=** 0 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testValidOralPracticalMark_2* method	*CheckOralPracticalMark* method Return true	The student *Oral Practical Mark* can be saved in the prase module in its data field	*CheckOralPracticalMark* method Return true As expected	y

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
17	Test correct OralPracticalMark	7	1	**mark=** 12 **fullMark**= 120	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testValidOralPracticalMark_3* method	*CheckOralPracticalMark* method Return true	The student *Oral Practical Mark* can be saved in the prase module in its data field	*CheckOralPracticalMark* method Return true As expected	y
18	Test incorrect OralPracticalMark	7	1	**mark=** -1 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInValidOralPracticalMark_1* method	*CheckOralPracticalMark* method Return false	The student *Oral Practical Mark* can't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckOralPracticalMark* method Return false As expected	y

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
19	Test incorrect OralPracticalMark	7	1	**mark**= 11 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInValidOralPracticalMark_2* method	*CheckOralPracticalMark* method Return false	The student *Oral Practical Mark* can't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckOralPracticalMark* method Return false As expected	y
20	Test incorrect OralPracticalMark	7	1	**mark**= 13 **fullMark**= 120	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInValidOralPracticalMark_3* method	*CheckOralPracticalMark* method Return false	The student *Oral Practical Mark* can't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckOralPracticalMark* method Return false As expected	y

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
21	Test correct MidtermExam mark	8	1	**mark=** 20 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testValidMidtermExamMark_1* method	*CheckMidtermExamMark* method Return true	The student *Midterm Exam Mark* can be saved in the prase module in its data field	*CheckMidtermExamMark* method Return true As expected	y
22	Test correct MidtermExam mark	8	1	**mark=** 0 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testValidMidtermExamMark_2* method	*CheckMidtermExamMark* method Return true	The student *Midterm Exam Mark* can be saved in the prase module in its data field	*CheckMidtermExamMark* method Return true As expected	y

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
23	Test incorrect MidtermExam mark	8	1	**mark=** -1 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInValidMidtermExamMark_1* method	*CheckMidtermExamMark* method Return false	The student *Midterm Exam Mar* kcan't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckMidtermExamMark* method Return false As expected	y
24	Test incorrect MidtermExam mark	8	1	**mark=** 21 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInValidMidtermExamMark_2* method	*CheckMidtermExamMark* method Return false	The student *Midterm Exam Mar* kcan't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckMidtermExamMark* method Return false As expected	y

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
25	Test correct FinalExam mark	9	1	**mark=** 60 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testValidFinalExamMark_1* method	*CheckFinalExamMark* method Return true	The student *final Exam Mark* can be saved in the prase module in its data field	*CheckFinalExamMark* method Return true As expected	y
26	Test correct FinalExam Mark	9	1	**mark=** 72 **fullMark**= 120	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testValidFinalExamMark_2* method	*CheckFinalExamMark* method Return true	The student *final Exam Mark* can be saved in the prase module in its data field	*CheckFinalExamMark* method Return true As expected	y

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
27	Test incorrect FinalExam mark	9	1	**mark=** -1 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInValidFinalExamMark_3* method	*CheckFinalExamMark* method Return false	The student *final Exam Mark* kcan't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckFinalExamMark* method Return false As expected	y
28	Test incorrect FinalExam mark	9	1	**mark=** 61 **fullMark**= 100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test of *testInValidFinalExamMark_4* method	*CheckFinalExamMark* method Return false	The student *final Exam Makr* kcan't be saved in the prase module in its data field and error message is printed in terminal showing the error location	*CheckFinalExamMark* method Return false As expected	y

Testing Type Details: Test Subject Details , Name and full mark
Tester Name: Anas Hamed
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/ Fail (Y/N)
1	Test Valid Student Name			“Anas Hamed”	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkName method return true	The Student name can be saved in the prase module in its data field	As expected	y
2	Test valid Course Name			“Math”	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkName method return true	The Course name can be saved in the prase module in its data field	checkName method return true as expected	y
3	Test invalid course name			" Testing"	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkName method return false	The course name cant be saved in its data field And error massage displayed on terminal	checkName method return false as expected	y

Testing Type Details: Test Subject Details , Name and full mark

Tester Name: Anas Hamed

Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/ Fail (Y/N)
4	Test Invalid Student Name			"" (empty)	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkName method return false	The student name cant be saved in its data field And error massage displayed	checkName method return false as expected	y
5	Test Invalid Course name			"Control2"	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkName method return false	The course name cant be saved in its data field And error massage displayed on terminal	checkName method return false as expected	y
6	Test Invalid Course name			"M@th2"	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkName method return false	The course name cant be saved in its data field And error massage displayed on terminal	checkName method return false as expected	y
7	Test valid Subject Code			"CSE321s"	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkSubjectCode method return true	The Course code can be saved in the prase module in its data field	checkSubjectCode method return true as expected	y

Testing Type Details: Test Subject Details , Name and full mark

Tester Name: Anas Hamed

Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/ Fail (Y/N)
8	Test valid Subject Code			"CSE345"	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkSubjectCode method return true	The Course code can be saved in the prase module in its data field	checkSubjectCode method return true as expected	y
9	Test Invalid Subject Code			"CSE31"	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkSubjectCode method return false	The Course code cant be saved in its data field And error massage displayed on terminal	checkSubjectCode method return false as expected	y
10	Test Invalid Subject Code			"CSE3131s"	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkSubjectCode method return false	The Course code cant be saved in its data field And error massage displayed on terminal	checkSubjectCode method return false as expected	y
11	Test Invalid Subject Code			“%SE321”	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkSubjectCode method return false	The Course code cant be saved in its data field And error massage displayed on terminal	checkSubjectCode method return false as expected	y

Testing Type Details: Test Subject Details , Name and full mark
Tester Name: Anas Hamed
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/ Fail (Y/N)
12	Test Invalid Subject Code			"S3E321"”	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkSubjectCode method return false	The Course code cant be saved in its data field And error massage displayed on terminal	checkSubjectCode method return false as expected	y
13	Test Invalid Subject Code			"CS 321s"	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkSubjectCode method return false	The Course code cant be saved in its data field And error massage displayed on terminal	checkSubjectCode method return false as expected	y
14	Test Invalid Subject Code			"CSE3 1"	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkSubjectCode method return false	The Course code cant be saved in its data field And error massage displayed on terminal	checkSubjectCode method return false as expected	y
15	Test Invalid Subject Code			"CSE3#1"	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkSubjectCode method return false	The Course code cant be saved in its data field And error massage displayed on terminal	checkSubjectCode method return false as expected	y

Testing Type Details: Test Subject Details , Name and full mark
Tester Name: Anas Hamed
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/ Fail (Y/N)
16	Test Invalid Subject Code			"CSE3a1"	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkSubjectCode method return false	The Course code cant be saved in its data field And error massage displayed on terminal	checkSubjectCode method return false as expected	y
17	Test Invalid Subject Code			"CSE331h"	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkSubjectCode method return false	The Course code cant be saved in its data field And error massage displayed on terminal	checkSubjectCode method return false as expected	y
18	Test Valid Full Mark			100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkFullMark method return true	The full mark can be saved in the prase module in its data field	checkFullMark method return true as expected	y
19	Test Invalid Full Mark			-100	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkFullMark method return false	The Full Mark cant be saved in its data field And error massage displayed on terminal	checkFullMark method return false as expected	y

Testing Type Details: Test Subject Details , Name and full mark
Tester Name: Anas Hamed
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/ Fail (Y/N)
20	Test Invalid Full Mark			54	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	checkFullMark method return false	The Full Mark cant be saved in its data field And error massage displayed on terminal	checkFullMark method return false as expected	y

Testing Type Details: Test reading input file
Tester Name: Ahmed Gamal
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
1	Test Open File With Valid File			Valid file: input_data.txt	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	Open file is succeeded	Data is produced as an array of strings and ready to be parsed by the parser module	As expected.	y
2	Test Open File With Invalid Extension			inValid file file extension: input_data.csv	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	Open file is failed	Data isn't produced as an array of strings and ready to be parsed by the parser module.	As expected.	y

Testing Type Details: Test reading input file
Tester Name: Ahmed Gamal
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
3	Test Open File With Empty Lines			Empty file : empty_file.txt	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	File is empty	Data is not available to be produced	As expected.	y
4	Test Open File With Non existent File			No file	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	No file was founded	Data is not available to be produced	As expected.	y

Testing Type Details: Test GPA Calculations and Grades
Tester Name: Mariam Ahmed
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
1	Test A+ grade			Activities:10 Oral/Practical:10 Midterm:19 Final:58	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	The expected grade is A+ with a GPA of 4.0.	The student grade and gpa can be saved in it's information to be outputted.	As expected.	y
2	Test A grade			Activities:9 Oral/Practical:10 Midterm:19 Final:55	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	The expected grade is A with a GPA of 4.0.	The student grade and gpa can be saved in it's information to be outputted.	As expected.	y

Testing Type Details: Test GPA Calculations and Grades
Tester Name: Mariam Ahmed
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
3	Test A- grade			Activities:8 Oral/Practical:10 Midterm:20 Final:51	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	The expected grade is A- with a GPA of 3.7.	The student grade and gpa can be saved in it's information to be outputted.	As expected.	y
4	Test B+ grade			Activities:9 Oral/Practical:9 Midterm:19 Final:47	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	The expected grade is A with a GPA of 3.3.	The student grade and gpa can be saved in it's information to be outputted.	As expected.	y

Testing Type Details: Test GPA Calculations and Grades
Tester Name: Mariam Ahmed
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
5	Test B grade.			Activities:10 Oral/Practical:10 Midterm:20 Final:40	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	The expected grade is B with a GPA of 3.0.	The student grade and gpa can be saved in it's information to be outputted.	As expected.	y
6	Test B- grade.			Activities:10 Oral/Practical:10 Midterm:20 Final:36	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	The expected grade is B- with a GPA of 2.7.	The student grade and gpa can be saved in it's information to be outputted.	As expected.	y

Testing Type Details: Test GPA Calculations and Grades
Tester Name: Mariam Ahmed
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
7	Test C+ grade			Activities:10 Oral/Practical:10 Midterm:20 Final:33	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	The expected grade is C+ with a GPA of 2.3.	The student grade and gpa can be saved in it's information to be outputted.	As expected.	y
8	Test C grade			Activities:10 Oral/Practical:10 Midterm:20 Final:30	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	The expected grade is C with a GPA of 2.0.	The student grade and gpa can be saved in it's information to be outputted.	As expected.	y

Testing Type Details: Test GPA Calculations and Grades
Tester Name: Mariam Ahmed
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
9	Test C- grade			Activities:10 Oral/Practical:10 Midterm:20 Final:27	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	The expected grade is C- with a GPA of 1.7.	The student grade and gpa can be saved in it's information to be outputted.	As expected.	y
10	Test D+ grade			Activities:10 Oral/Practical:10 Midterm:20 Final:24	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	The expected grade is D+ with a GPA of 1.3.	The student grade and gpa can be saved in it's information to be outputted.	As expected.	y

Testing Type Details: Test GPA Calculations and Grades
Tester Name: Mariam Ahmed
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
11	Test D grade			Activities:10 Oral/Practical:10 Midterm:20 Final:20	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	The expected grade is D with a GPA of 1.0.	The student grade and gpa can be saved in it's information to be outputted.	As expected.	y
12	Test F grade			Activities:9 Oral/Practical:5 Midterm:14 Final:30	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	The expected grade is F with a GPA of 0.0.	The student grade and gpa can be saved in it's information to be outputted.	As expected.	y

Testing Type Details: Test GPA Calculations and Grades
Tester Name: Mariam Ahmed
Test Environment Details

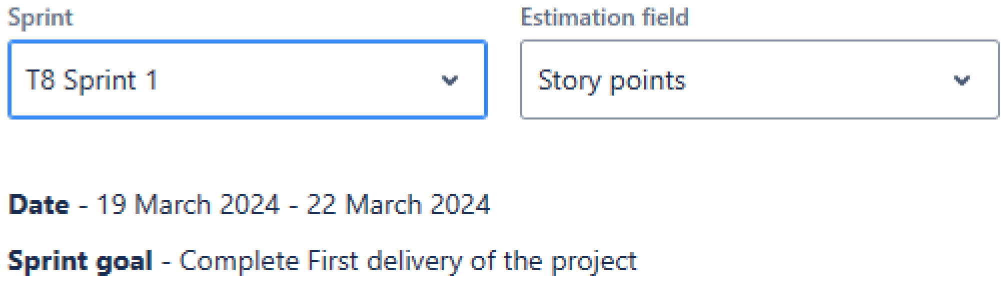
Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
13	Invalid input where the total mark is greater than 100.			Activities:10 Oral/Practical:10 Midterm:20 Final:110	1.Create test method 2.Create test data 3. Use assertions to verify that an IllegalArgumentException is thrown. 4.run a junit test	An IllegalArgumentException should be thrown.	The student grade and gpa can not be saved in it's information and a message "Invalid total mark" is shown.	As expected.	y
14	Invalid input where the total mark is less than 0.			Activities:10 Oral/Practical:10 Midterm:20 Final:-110	1.Create test method 2.Create test data 3. Use assertions to verify that an IllegalArgumentException is thrown. 4.run a junit test	An IllegalArgumentException should be thrown.	The student grade and gpa can not be saved in it's information and a message "Invalid total mark" is shown.	As expected.	y

Testing Type Details: Test output file function
Tester Name: Sara Ashraf
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
1	File made scenerio			There is subject object with all its data	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	"Text file created successfully" is printed	File is created and data inside it as required	As expected.	y
2	File not made scenerio			No subject object(null)	1.Create test method 2.Create test data 3.use assertion to compare the expected and the actual results 4.run a junit test	"no file is created" is printed	File is not created	As expected.	y

Jira ScreenShots

Sprint 1 reports:



Sprint: T8 Sprint 1

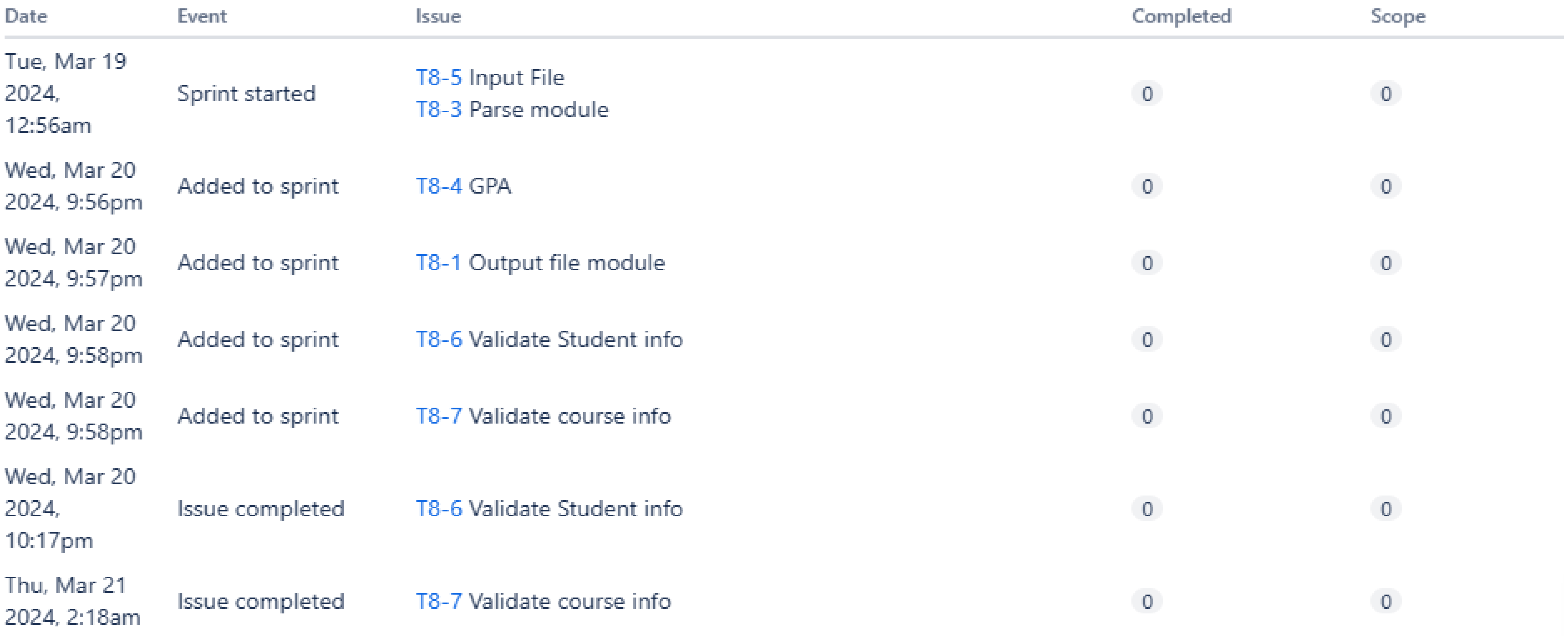
Estimation field: Story points

Date - 19 March 2024 - 22 March 2024

Sprint goal - Complete First delivery of the project

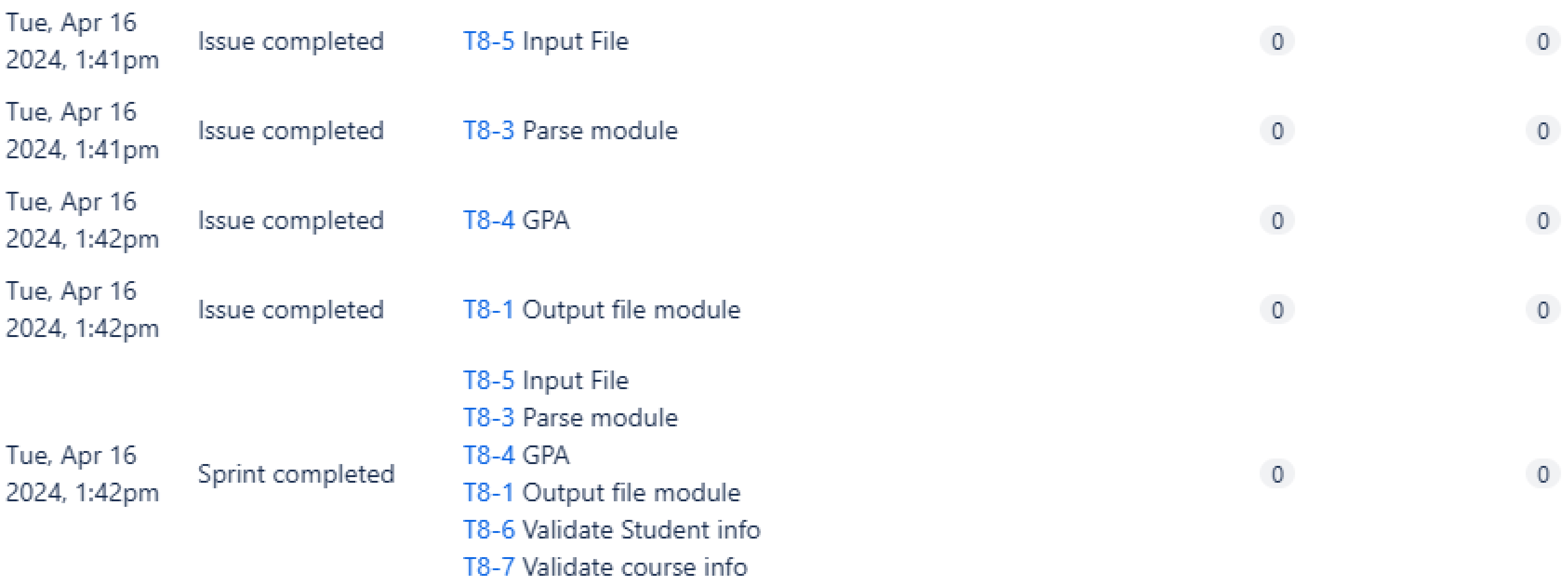
Jira ScreenShots

Burnup report:



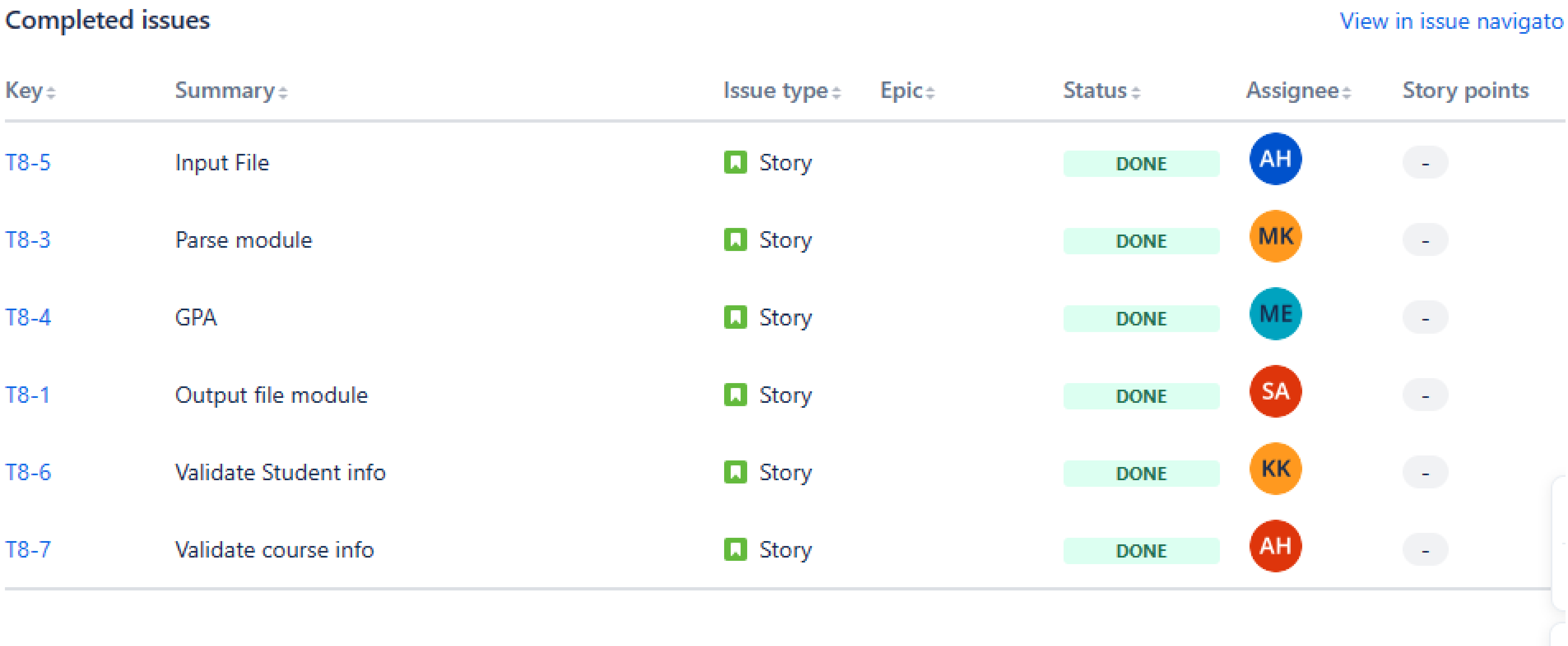
Date	Event	Issue	Completed	Scope
Tue, Mar 19 2024, 12:56am	Sprint started	T8-5 Input File T8-3 Parse module	0	0
Wed, Mar 20 2024, 9:56pm	Added to sprint	T8-4 GPA	0	0
Wed, Mar 20 2024, 9:57pm	Added to sprint	T8-1 Output file module	0	0
Wed, Mar 20 2024, 9:58pm	Added to sprint	T8-6 Validate Student info	0	0
Wed, Mar 20 2024, 9:58pm	Added to sprint	T8-7 Validate course info	0	0
Wed, Mar 20 2024, 10:17pm	Issue completed	T8-6 Validate Student info	0	0
Thu, Mar 21 2024, 2:18am	Issue completed	T8-7 Validate course info	0	0

Jira ScreenShots



Tue, Apr 16 2024, 1:41pm	Issue completed	T8-5 Input File	0	0
Tue, Apr 16 2024, 1:41pm	Issue completed	T8-3 Parse module	0	0
Tue, Apr 16 2024, 1:42pm	Issue completed	T8-4 GPA	0	0
Tue, Apr 16 2024, 1:42pm	Issue completed	T8-1 Output file module	0	0
Tue, Apr 16 2024, 1:42pm	Sprint completed	T8-5 Input File T8-3 Parse module T8-4 GPA T8-1 Output file module T8-6 Validate Student info T8-7 Validate course info	0	0

Jira ScreenShots

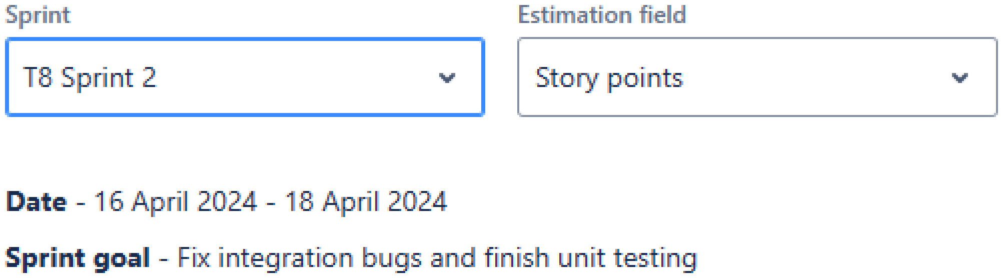


Completed issues

Key	Summary	Issue type	Epic	Status	Assignee	Story points
T8-5	Input File	Story		DONE	AH	-
T8-3	Parse module	Story		DONE	MK	-
T8-4	GPA	Story		DONE	ME	-
T8-1	Output file module	Story		DONE	SA	-
T8-6	Validate Student info	Story		DONE	KK	-
T8-7	Validate course info	Story		DONE	AH	-

Jira ScreenShots

Sprint 2 reports:



Sprint

T8 Sprint 2

Estimation field

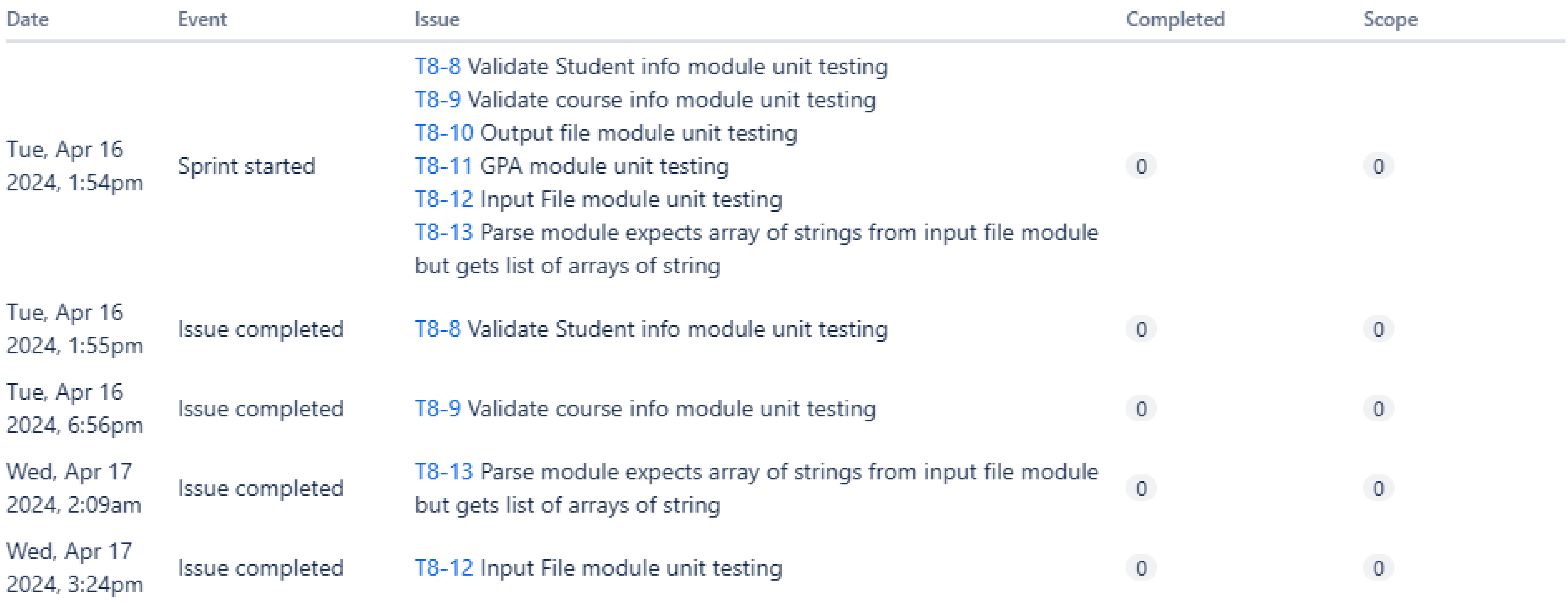
Story points

Date - 16 April 2024 - 18 April 2024

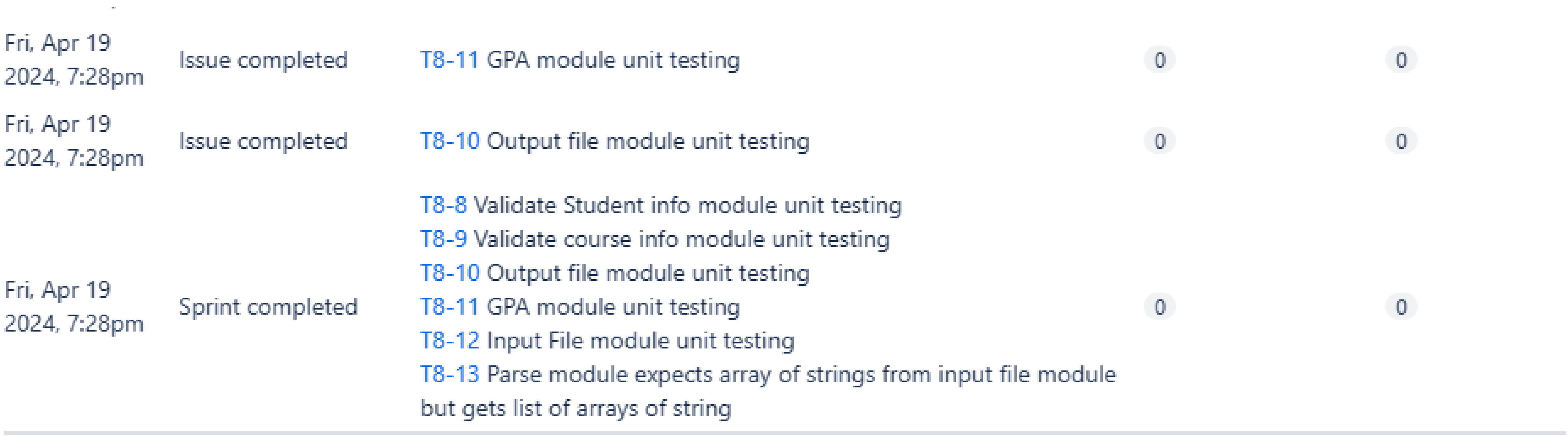
Sprint goal - Fix integration bugs and finish unit testing

Jira ScreenShots

Burnup report:



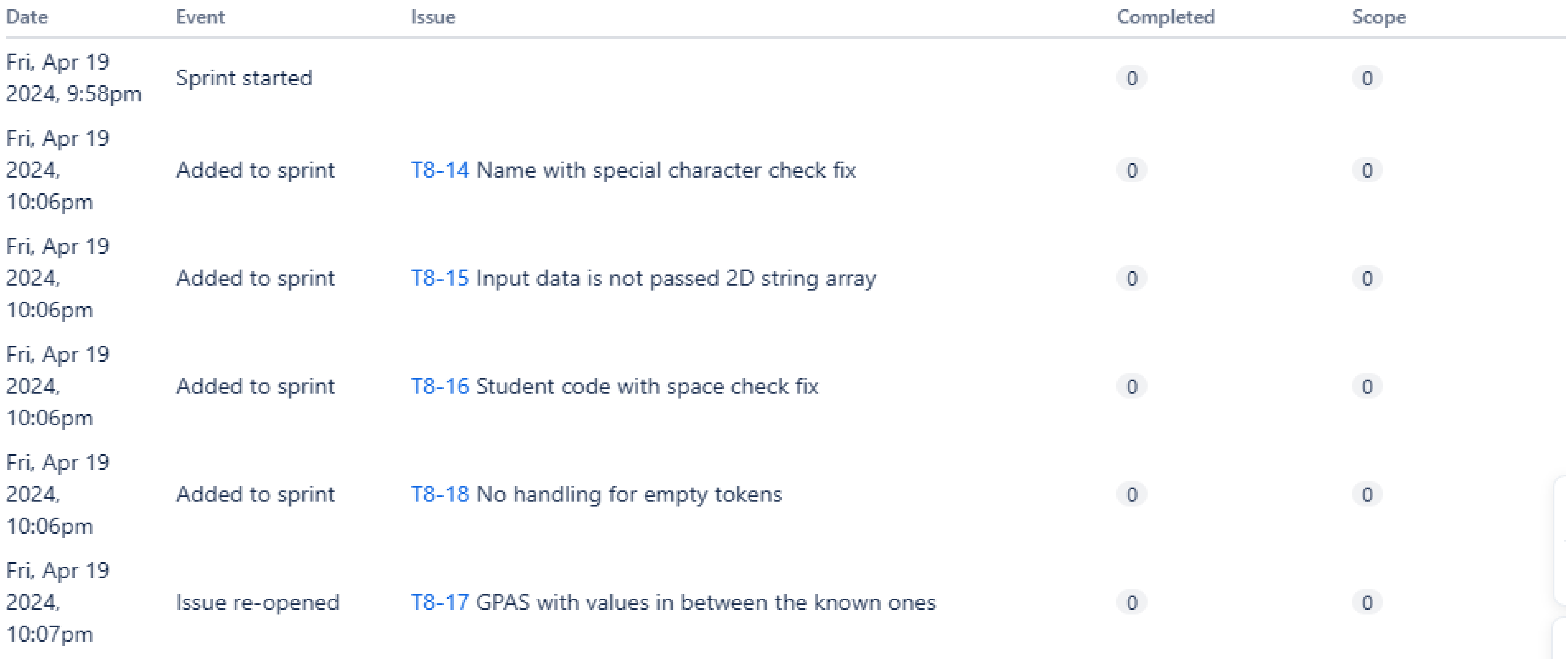
Date	Event	Issue	Completed	Scope
Tue, Apr 16 2024, 1:54pm	Sprint started	T8-8 Validate Student info module unit testing T8-9 Validate course info module unit testing T8-10 Output file module unit testing T8-11 GPA module unit testing T8-12 Input File module unit testing T8-13 Parse module expects array of strings from input file module but gets list of arrays of string	0	0
Tue, Apr 16 2024, 1:55pm	Issue completed	T8-8 Validate Student info module unit testing	0	0
Tue, Apr 16 2024, 6:56pm	Issue completed	T8-9 Validate course info module unit testing	0	0
Wed, Apr 17 2024, 2:09am	Issue completed	T8-13 Parse module expects array of strings from input file module but gets list of arrays of string	0	0
Wed, Apr 17 2024, 3:24pm	Issue completed	T8-12 Input File module unit testing	0	0



Fri, Apr 19 2024, 7:28pm	Issue completed	T8-11 GPA module unit testing	0	0
Fri, Apr 19 2024, 7:28pm	Issue completed	T8-10 Output file module unit testing	0	0
Fri, Apr 19 2024, 7:28pm	Sprint completed	T8-8 Validate Student info module unit testing T8-9 Validate course info module unit testing T8-10 Output file module unit testing T8-11 GPA module unit testing T8-12 Input File module unit testing T8-13 Parse module expects array of strings from input file module but gets list of arrays of string	0	0

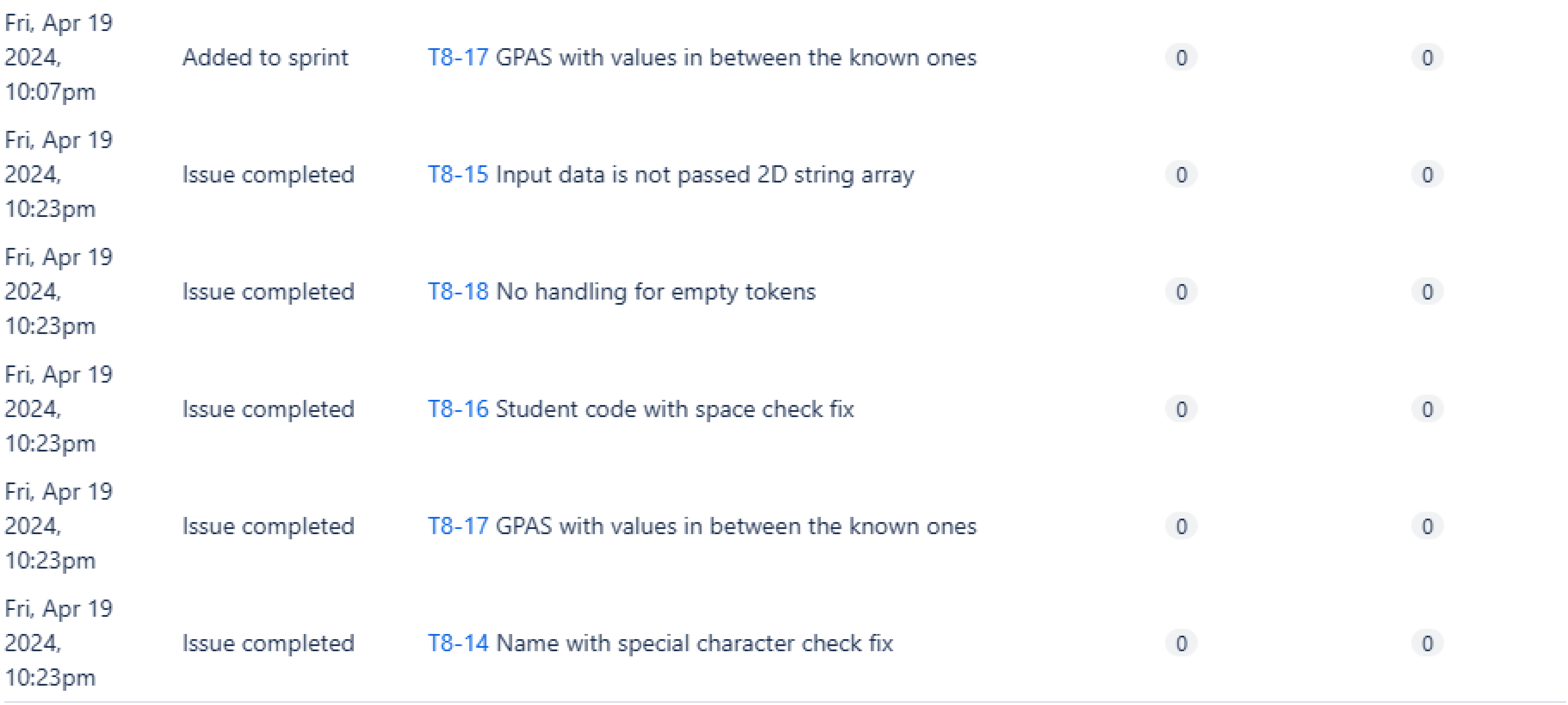
Jira ScreenShots

Sprint 3 reports:
Burnup report:



Date	Event	Issue	Completed	Scope
Fri, Apr 19 2024, 9:58pm	Sprint started		0	0
Fri, Apr 19 2024, 10:06pm	Added to sprint	T8-14 Name with special character check fix	0	0
Fri, Apr 19 2024, 10:06pm	Added to sprint	T8-15 Input data is not passed 2D string array	0	0
Fri, Apr 19 2024, 10:06pm	Added to sprint	T8-16 Student code with space check fix	0	0
Fri, Apr 19 2024, 10:06pm	Added to sprint	T8-18 No handling for empty tokens	0	0
Fri, Apr 19 2024, 10:07pm	Issue re-opened	T8-17 GPAS with values in between the known ones	0	0

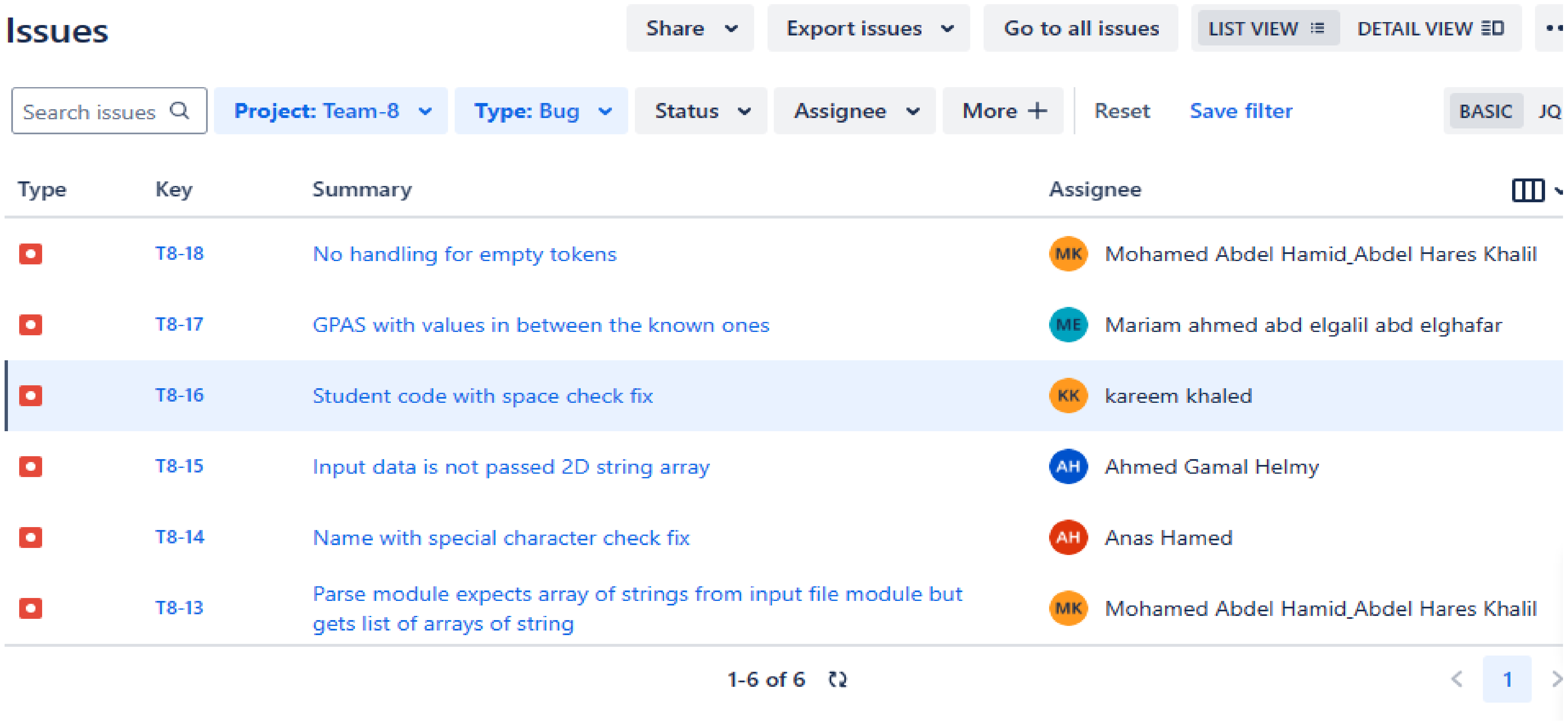
Jira ScreenShots



Fri, Apr 19 2024, 10:07pm	Added to sprint	T8-17 GPAS with values in between the known ones	0	0
Fri, Apr 19 2024, 10:23pm	Issue completed	T8-15 Input data is not passed 2D string array	0	0
Fri, Apr 19 2024, 10:23pm	Issue completed	T8-18 No handling for empty tokens	0	0
Fri, Apr 19 2024, 10:23pm	Issue completed	T8-16 Student code with space check fix	0	0
Fri, Apr 19 2024, 10:23pm	Issue completed	T8-17 GPAS with values in between the known ones	0	0
Fri, Apr 19 2024, 10:23pm	Issue completed	T8-14 Name with special character check fix	0	0

Jira ScreenShots

Bugs:



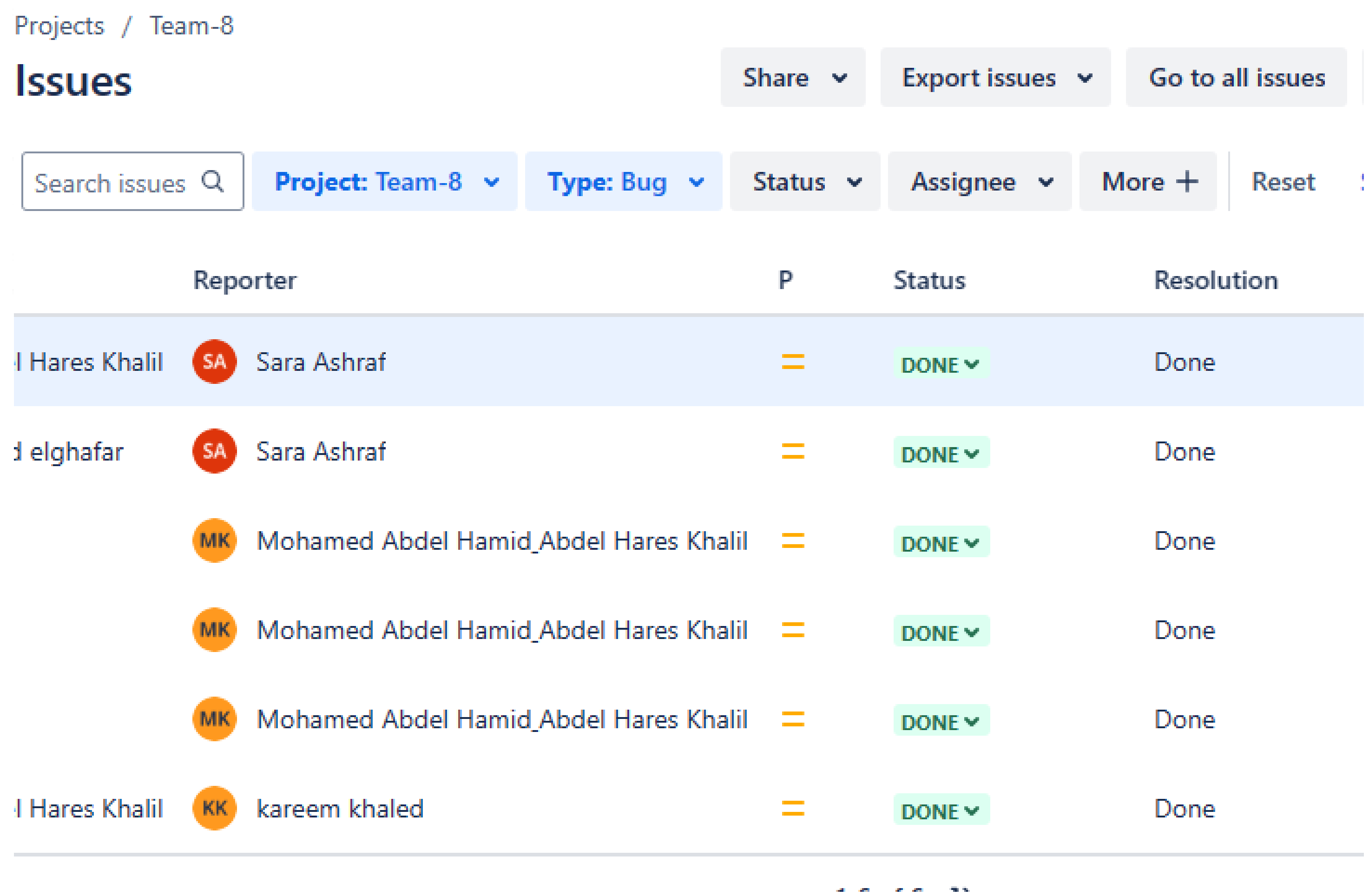
Project: Team-8

Type: Bug

Type	Key	Summary	Assignee
🟥	T8-18	No handling for empty tokens	MK Mohamed Abdel Hamid_Abdel Hares Khalil
🟥	T8-17	GPAS with values in between the known ones	ME Mariam ahmed abd elgalil abd elghafar
🟥	T8-16	Student code with space check fix	KK kareem khaled
🟥	T8-15	Input data is not passed 2D string array	AH Ahmed Gamal Helmy
🟥	T8-14	Name with special character check fix	AH Anas Hamed
🟥	T8-13	Parse module expects array of strings from input file module but gets list of arrays of string	MK Mohamed Abdel Hamid_Abdel Hares Khalil

1-6 of 6

Jira ScreenShots



	Reporter	P	Status	Resolution
Hares Khalil	SA Sara Ashraf	=	DONE	Done
elghafar	SA Sara Ashraf	=	DONE	Done
	MK Mohamed Abdel Hamid_Abdel Hares Khalil	=	DONE	Done
	MK Mohamed Abdel Hamid_Abdel Hares Khalil	=	DONE	Done
	MK Mohamed Abdel Hamid_Abdel Hares Khalil	=	DONE	Done
Hares Khalil	KK kareem khaled	=	DONE	Done

Integration Testing: Bottom-Up Approach

The Bottom-up approach is carried out by running the bottom module test cases which are:
From T1 to T28 of DataCheckModule
And from T1 to T20 of Test Subject Details , Name and full mark module

And running the top module test cases which are:
From T1 to T16 of Test Parse File Module

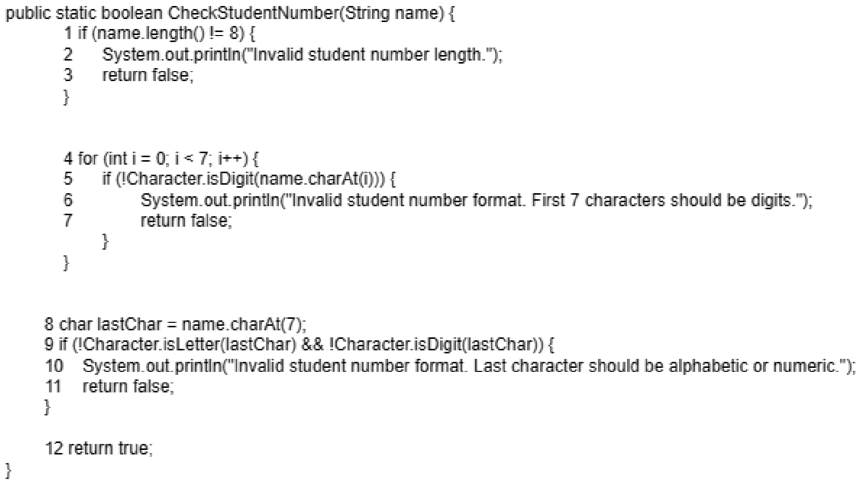
Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

White Box Testing:

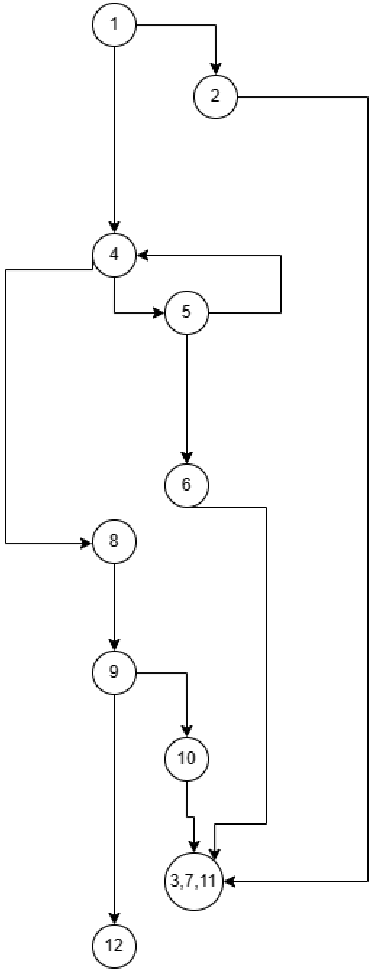
E – Number of edges = 13
N – Number of Nodes = 11
V(G) = E - N + 2 =13-11+2=4

Path	Test Cases
1,4,5,4,8,9,12	#1 ,#2
1,4,5,6,7	#7
1,2,3	#3
1,4,5,4,8,9,10,11	#8

Number of basis paths =4
Number of covered basis paths = 4
So basis path coverage 100%
As Basis Path coverage implies Branch coverage So Branch coverage 100%
As Branch coverage implies statement coverage so Statement coverage 100%



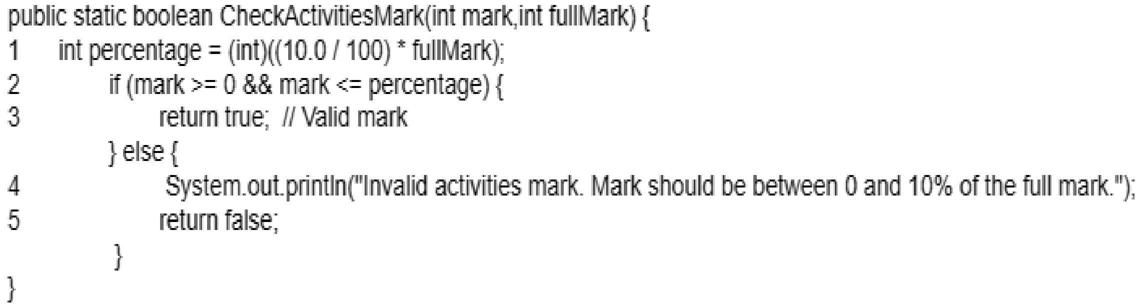
```
public static boolean CheckStudentNumber(String name) {
        1 if (name.length() != 8) {
        2     System.out.println("Invalid student number length.");
        3     return false;
        }


        4 for (int i = 0; i < 7; i++) {
        5     if (!Character.isDigit(name.charAt(i))) {
        6             System.out.println("Invalid student number format. First 7 characters should be digits.");
        7             return false;
                }
        }


      8 char lastChar = name.charAt(7);
      9 if (!Character.isLetter(lastChar) && !Character.isDigit(lastChar)) {
      10   System.out.println("Invalid student number format. Last character should be alphabetic or numeric.");
      11   return false;
      }

      12 return true;
}
```



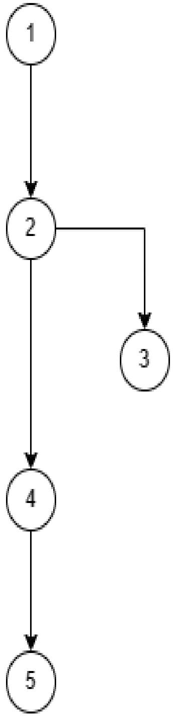
Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

E – Number of edges = 4
N – Number of Nodes =5
V(G) = E - N + 2 =4-5+2=1

Path	Test Cases
1,2,4,5	#12,#13,#14
1,2,3	#9,#10,#11



```
public static boolean CheckActivitiesMark(int mark,int fullMark) {
1     int percentage = (int)((10.0 / 100) * fullMark);
2           if (mark >= 0 && mark <= percentage) {
3                 return true;  // Valid mark
            } else {
4                 System.out.println("Invalid activities mark. Mark should be between 0 and 10% of the full mark.");
5                 return false;
            }
}
```



Number of full paths =2

Number of covered full paths = 2

So full path coverage 100%

As Basis Path coverage implies basis path coverage

So basis path coverage 100%

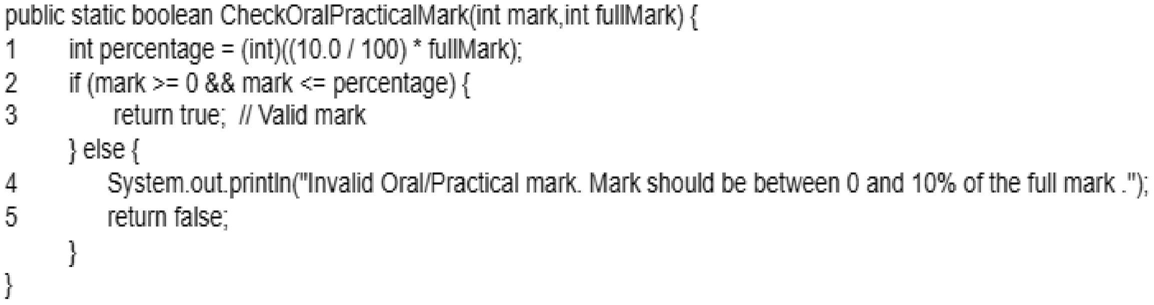
As Basis Path coverage implies Branch coverage So Branch coverage 100%

As Branch coverage implies statement coverage so Statement coverage 100%

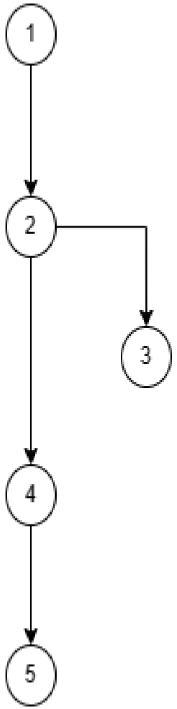
Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

E – Number of edges = 4
N – Number of Nodes =5
V(G) = E - N + 2 =4-5+2=1

Path	Test Cases
1,2,4,5	#15,#16,#17
1,2,3	#18,#19,#20



```
public static boolean CheckOralPracticalMark(int mark,int fullMark) {
1       int percentage = (int)((10.0 / 100) * fullMark);
2       if (mark >= 0 && mark <= percentage) {
3           return true;  // Valid mark
        } else {
4           System.out.println("Invalid Oral/Practical mark. Mark should be between 0 and 10% of the full mark .");
5           return false;
        }
}
```



Number of full paths =2

Number of covered full paths = 2

So full path coverage 100%

As Basis Path coverage implies basis path coverage

So basis path coverage 100%

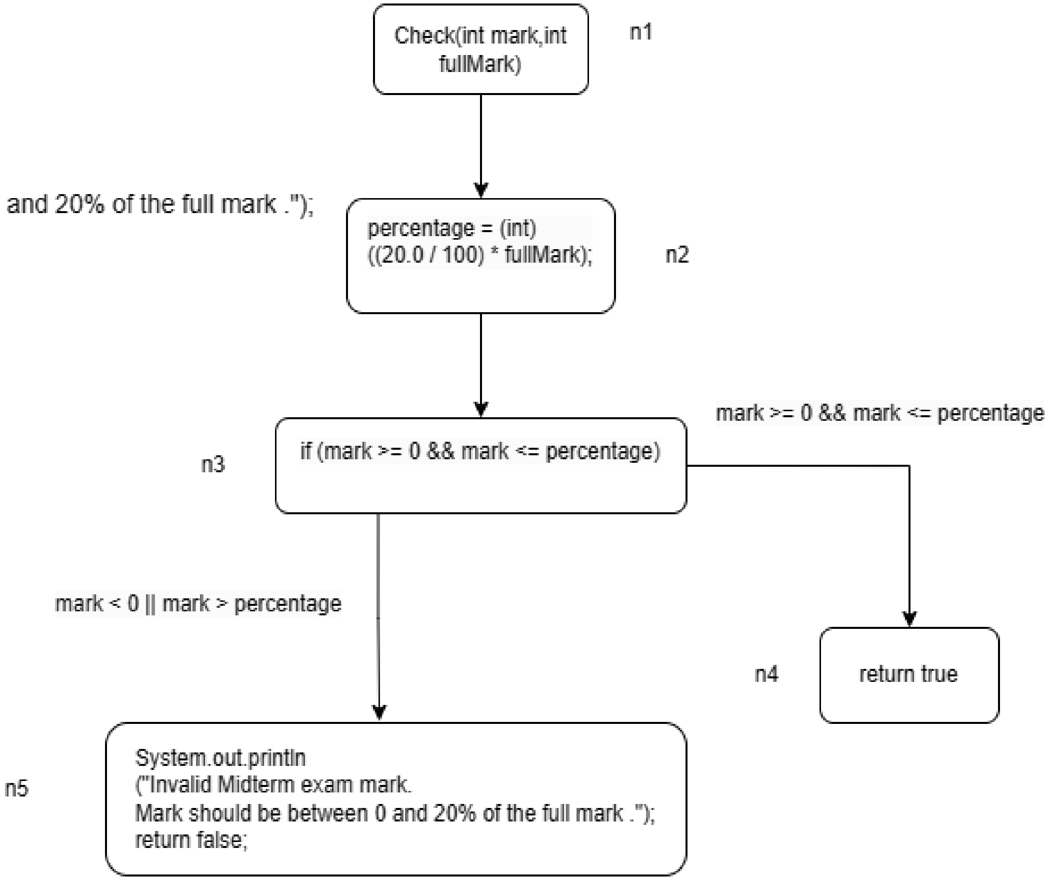
As Basis Path coverage implies Branch coverage So Branch coverage 100%

As Branch coverage implies statement coverage so Statement coverage 100%

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Data Flow Testing:

```
1 public static boolean CheckMidtermExamMark(int mark,int fullMark) {
2         int percentage = (int)((20.0 / 100) * fullMark);
3         if (mark >= 0 && mark <= percentage) {
4             return true;
          } else {
5             System.out.println("Invalid Midterm exam mark. Mark should be between 0 and 20% of the full mark .");
6             return false;
          }
  }
```


Check(int mark,int fullMark)
n1
percentage = (int) ((20.0 / 100) * fullMark);
n2
mark >= 0 && mark <= percentage
n3
if (mark >= 0 && mark <= percentage)
mark < 0 || mark > percentage
n4
return true
n5
System.out.println ("Invalid Midterm exam mark. Mark should be between 0 and 20% of the full mark ."); return false;

Identifying DU-Pairs Variable Mark:

Du-pair	paths
(1,<3,5>)	<1,2,3,5>
(1,<3,4>)	<1,2,3,4>

Identifying DU-PairsVariable fullMark:

Du-pair	paths
(1,2)	<1,2>

Identifying DU-PairsVariable percentage:

Du-pair	paths
(2,<3,5>)	<1,2,3,5>
(2,<3,4>)	<1,2,3,4>

T21:<1,2,3,4>
T23:<1,2,3,5>

Both test cases together provide All-uses coverage for all 3 Variables Mark, fullMark, percentage

Testing Type Details: DataCheckModule
Tester Name:Kareem Khaled
Test Environment Details

Identifying DU-Pairs
Variable name:

Du-pair	paths
(1,<2,3>)	<1,2,3>
(1,<2,4>)	<1,2,4>
(1,<5,6>)	<1,2,4,5,6>
(1,<5,4>)	<1,2,4,5,4>
(1,7)	<1,2,4,5,4,7>

Identifying DU-Pairs
Variable lastChar:

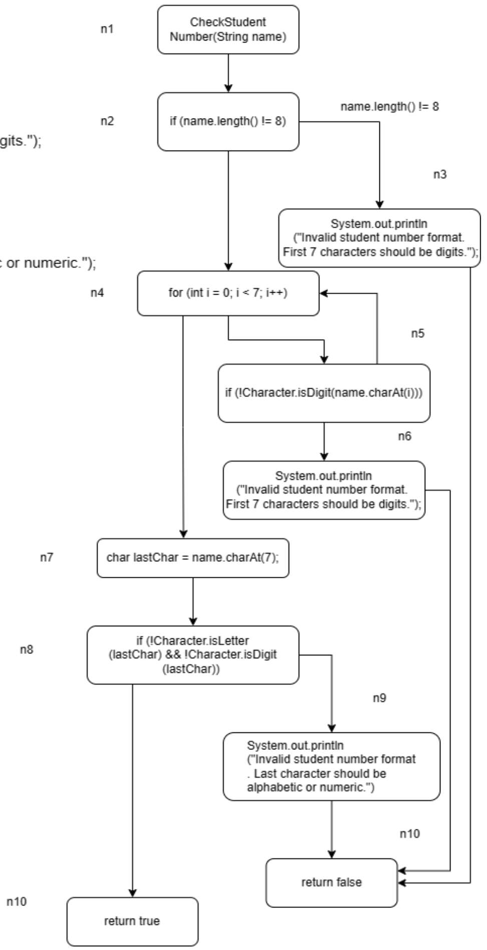
Du-pair	paths
(7,<8,9>)	<1,2,4,5,4,7,8,9>
(7,<8,10>)	<1,2,4,5,4,7,8,10>

T1:<1,2,4,5,4,7,8,10>
T3:<1,2,3,10>
T7:<1,2,4,5,6,10>
T8:<1,2,4,5,4,7,8,9,10>

All these four test cases together provide All-uses coverage both variables name and lastChar

϶

```
1 public static boolean CheckStudentNumber(String name) {
2           if (name.length() != 8) {
3                 System.out.println("Invalid student number length.");
4                 return false;
            }

5           for (int i = 0; i < 7; i++) {
6                 if (!Character.isDigit(name.charAt(i))) {
7                       System.out.println("Invalid student number format. First 7 characters should be digits.");
8                       return false;
                  }
            }

9           char lastChar = name.charAt(7);
10          if (!Character.isLetter(lastChar) && !Character.isDigit(lastChar)) {
11                System.out.println("Invalid student number format. Last character should be alphabetic or numeric.");
12                return false;
            }

13          return true;
      }
```


n1
CheckStudent Number(String name)
n2
if (name.length() != 8)
name.length() != 8
n3
System.out.println ("Invalid student number format. First 7 characters should be digits.");
n4
for (int i = 0; i < 7; i++)
n5
if (!Character.isDigit(name.charAt(i)))
n6
System.out.println ("Invalid student number format. First 7 characters should be digits.");
n7
char lastChar = name.charAt(7);
n8
if (!Character.isLetter (lastChar) && !Character.isDigit (lastChar))
n9
System.out.println ("Invalid student number format . Last character should be alphabetic or numeric.")
n10
return false
n10
return true

Testing Type Details: Test Subject Details , Name and full mark
Tester Name: Anas Hamed
Test Environment Details

White Box Testing:

(Basis path coverage)

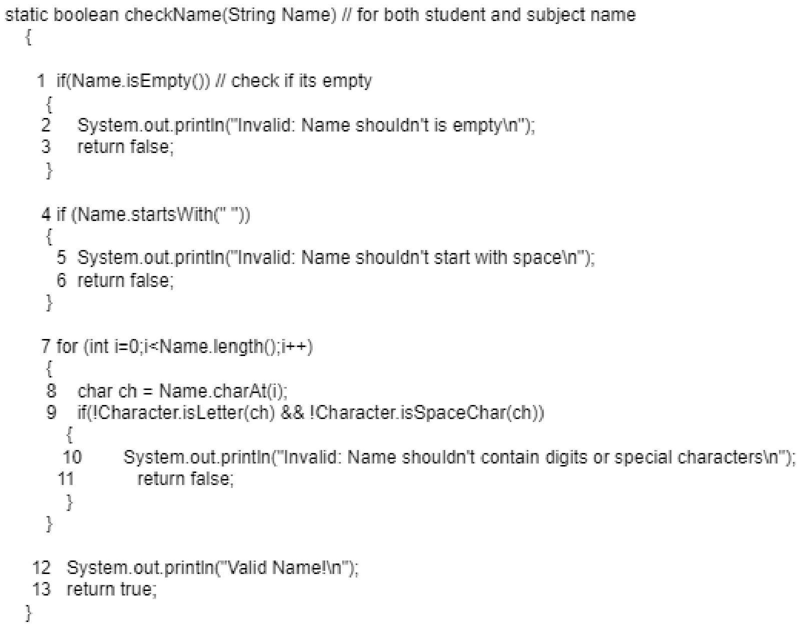
E – Number of edges = 13
N – Number of Nodes = 11
V(G) = E - N + 2 =13-11+2=4

Path	Test cases
1 -> 4 -> 7 -> 8 -> 9 -> 7 -> 12-> 13	#1 ,#2
1->2->3	#4
1->4->7->8->9->10->11	#6
1->4->5->11	#3

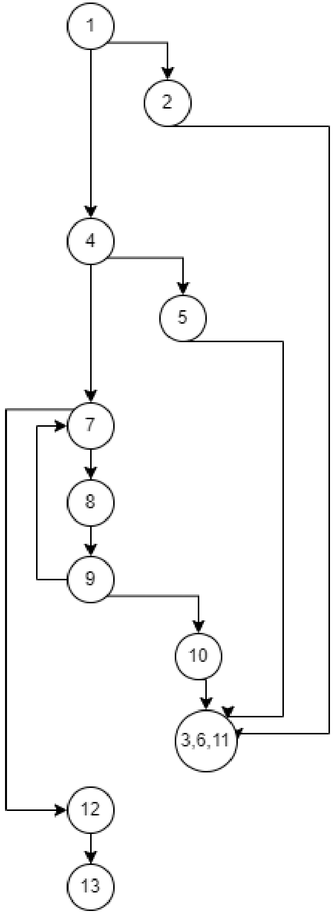
Number of basis paths =4
Number of covered basis paths = 4
So basis path coverage 100%

As Basis Path coverage implies Branch coverage
So Branch coverage 100%

As Branch coverage implies statement coverage so Statement coverage 100%



```
static boolean checkName(String Name) // for both student and subject name
  {

   1  if(Name.isEmpty()) // check if its empty
    {
    2    System.out.println("Invalid: Name shouldn't is empty\n");
    3    return false;
    }

   4 if (Name.startsWith(" "))
    {
      5  System.out.println("Invalid: Name shouldn't start with space\n");
      6  return false;
    }

   7 for (int i=0;i<Name.length();i++)
    {
    8    char ch = Name.charAt(i);
    9    if(!Character.isLetter(ch) && !Character.isSpaceChar(ch))
        {
      10        System.out.println("Invalid: Name shouldn't contain digits or special characters\n");
     11           return false;
        }
    }

  12   System.out.println("Valid Name!\n");
  13   return true;
  }
```



Testing Type Details: Test Subject Details , Name and full mark
Tester Name: Anas Hamed
Test Environment Details

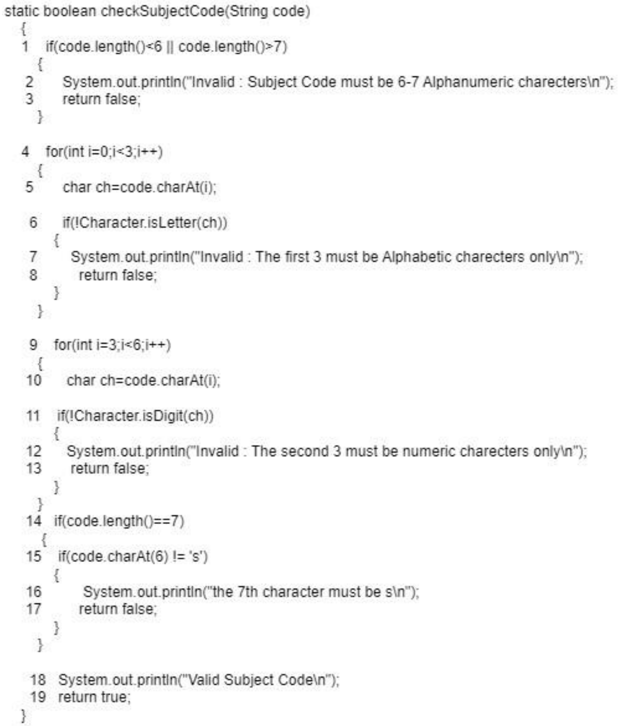
E – Number of edges = 22
N – Number of Nodes = 17
V(G) = E - N + 2 =22-17+2=7

Path	Test cases
1->4->5->6->4->9->10->11->9->14->18->19	#8
1->4->5->6->4->9->10->11->9->14->15->18->19	#7
1->2->3	#9
1->4->5->6->7->8	#11
1->4->5->6->4->5->6->7->8	#12,#13
1->4->5->6->4->9->10->11->9->10->11->12->13	#14,#15,#16
1->4->5->6->4->9->10->11->9->14->15->16->17	#17

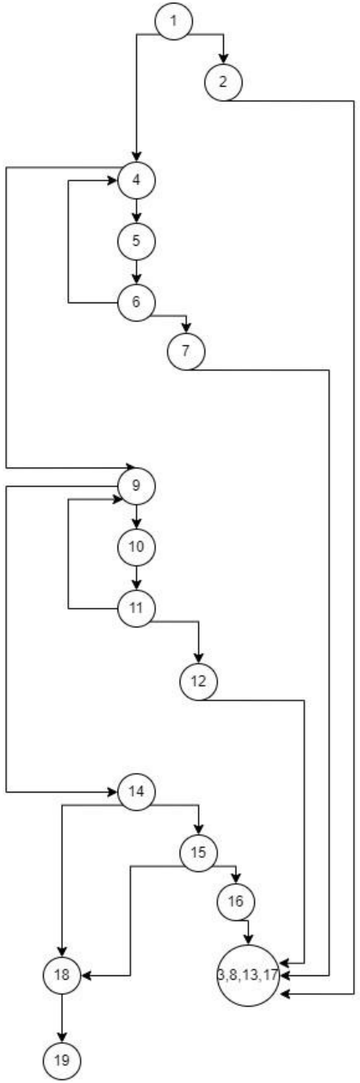
Number of basis paths =7
Number of covered basis paths = 7
So basis path coverage 100%

As Basis Path coverage implies Branch coverage
So Branch coverage 100%

As Branch coverage implies statement coverage
so Statement coverage 100%



```
static boolean checkSubjectCode(String code)
  {
  1   if(code.length()<6 || code.length()>7)
     {
   2     System.out.println("Invalid : Subject Code must be 6-7 Alphanumeric charecters\n");
   3     return false;
     }

  4   for(int i=0;i<3;i++)
     {
   5     char ch=code.charAt(i);

   6     if(!Character.isLetter(ch))
        {
   7      System.out.println("Invalid : The first 3 must be Alphabetic charecters only\n");
   8        return false;
        }
     }

   9  for(int i=3;i<6;i++)
    {
  10    char ch=code.charAt(i);

  11   if(!Character.isDigit(ch))
      {
  12    System.out.println("Invalid : The second 3 must be numeric charecters only\n");
  13     return false;
      }
    }
  14  if(code.length()==7)
    {
  15   if(code.charAt(6) != 's')
      {
  16      System.out.println("the 7th character must be s\n");
  17     return false;
      }
    }

  18  System.out.println("Valid Subject Code\n");
  19  return true;
 }
```



Testing Type Details: Test Parse File Module
Tester Name: Mohamed Abdel Hamid
Test Environment Details

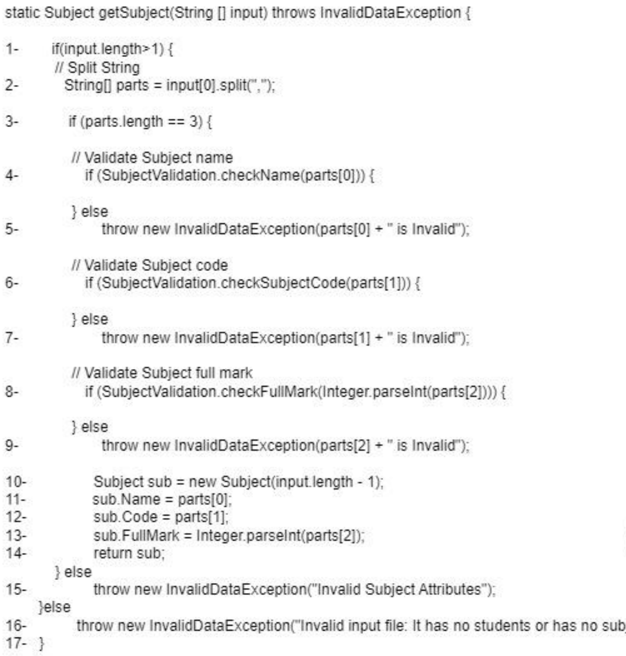
White Box Testing – Parse File Module

Basis path coverage for getSubjectMethod()

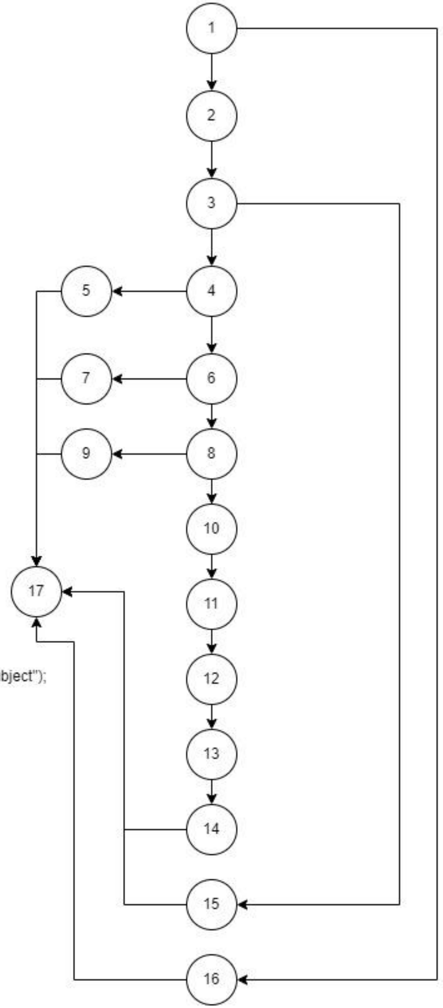
E – Number of edges = 22
N – Number of Nodes = 17
V(G) = E - N + 2 = 22 – 17 + 2 = 7

Path	Test cases
1 -> 16 -> 17	#2
1->2->3->4->6->8->10->12->13->14->17	#8
1->2->3->15->17	#1
1->2->3->4->5->17	#3
1->2->3->4->6->7->17	#4
1->2->3->4->6->8->9->17	#5

Number of basis paths = 7
Number of covered basis paths = 7
Basis path coverage = 100%
As Basis Path coverage implies Branch coverage => Branch coverage 100%
As Branch coverage implies statement coverage => Statement coverage 100%



```
static Subject getSubject(String [] input) throws InvalidDataException {

1-      if(input.length>1) {
         // Split String
2-         String[] parts = input[0].split(",");

3-          if (parts.length == 3) {

            // Validate Subject name
4-              if (SubjectValidation.checkName(parts[0])) {

            } else
5-                  throw new InvalidDataException(parts[0] + " is Invalid");

            // Validate Subject code
6-              if (SubjectValidation.checkSubjectCode(parts[1])) {

            } else
7-                  throw new InvalidDataException(parts[1] + " is Invalid");

            // Validate Subject full mark
8-              if (SubjectValidation.checkFullMark(Integer.parseInt(parts[2]))) {

            } else
9-                  throw new InvalidDataException(parts[2] + " is Invalid");

10-                Subject sub = new Subject(input.length - 1);
11-                sub.Name = parts[0];
12-                sub.Code = parts[1];
13-                sub.FullMark = Integer.parseInt(parts[2]);
14-                return sub;
          } else
15-                throw new InvalidDataException("Invalid Subject Attributes");
       }else
16-          throw new InvalidDataException("Invalid input file: It has no students or has no subject");
17-  }
```

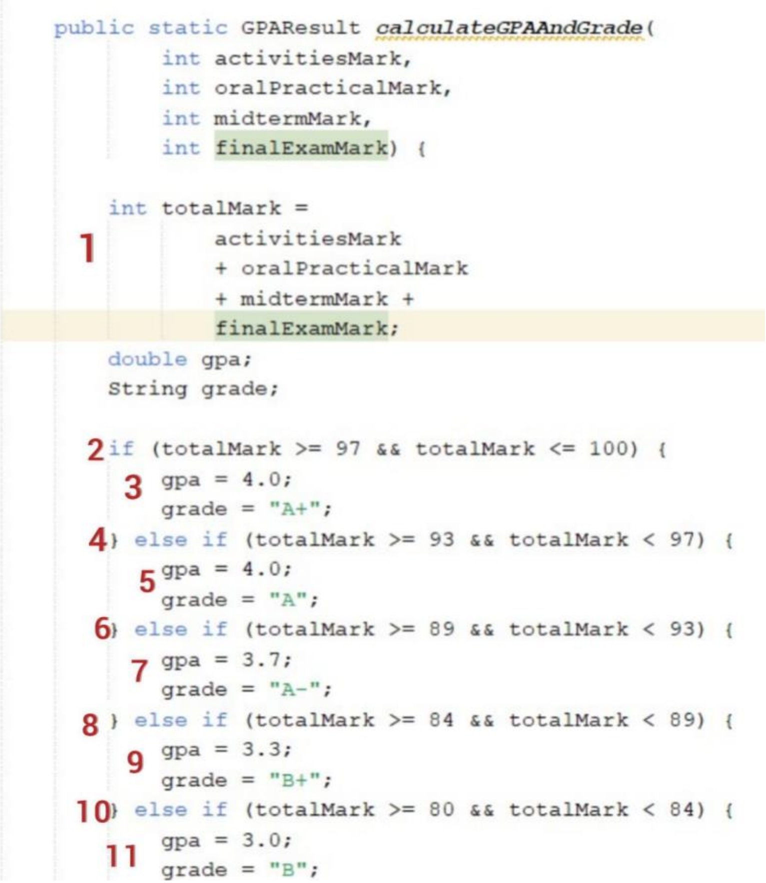


Testing Type Details: Test GPA Calculations and Grades

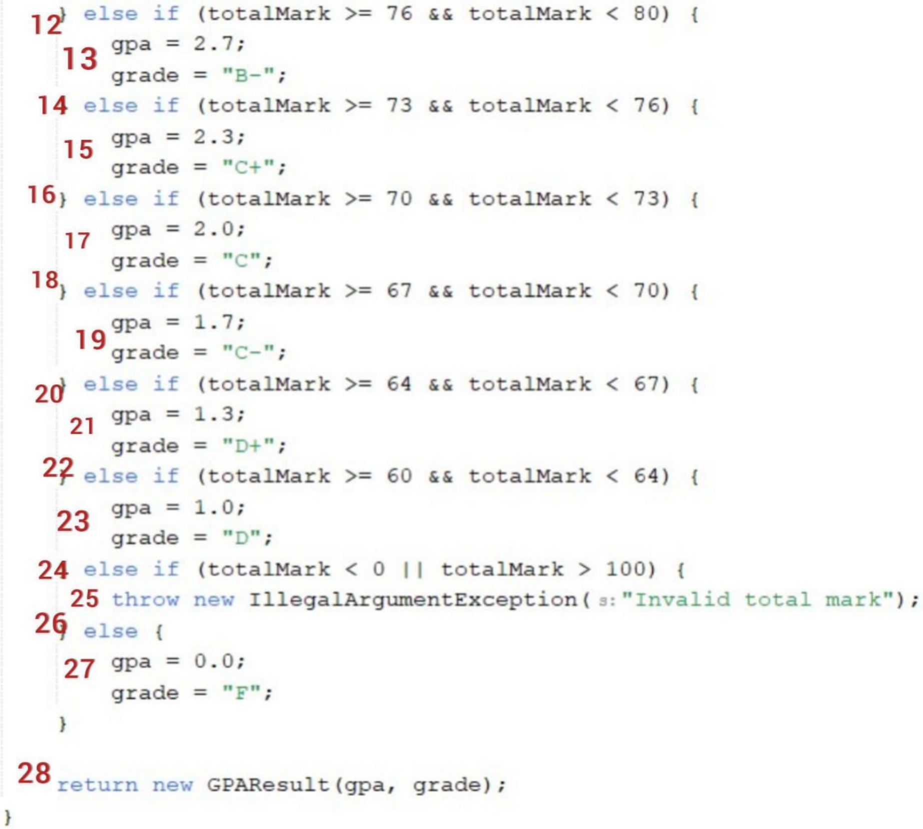
Tester Name: Mariam Ahmed

Test Environment Details

White Box Testing:



```
public static GPAResult calculateGPAAndGrade(
        int activitiesMark,
        int oralPracticalMark,
        int midtermMark,
        int finalExamMark) {

    int totalMark =
1           activitiesMark
            + oralPracticalMark
            + midtermMark +
            finalExamMark;
    double gpa;
    String grade;

2   if (totalMark >= 97 && totalMark <= 100) {
3       gpa = 4.0;
        grade = "A+";
4   } else if (totalMark >= 93 && totalMark < 97) {
5       gpa = 4.0;
        grade = "A";
6   } else if (totalMark >= 89 && totalMark < 93) {
7       gpa = 3.7;
        grade = "A-";
8   } else if (totalMark >= 84 && totalMark < 89) {
9       gpa = 3.3;
        grade = "B+";
10  } else if (totalMark >= 80 && totalMark < 84) {
11      gpa = 3.0;
        grade = "B";
```

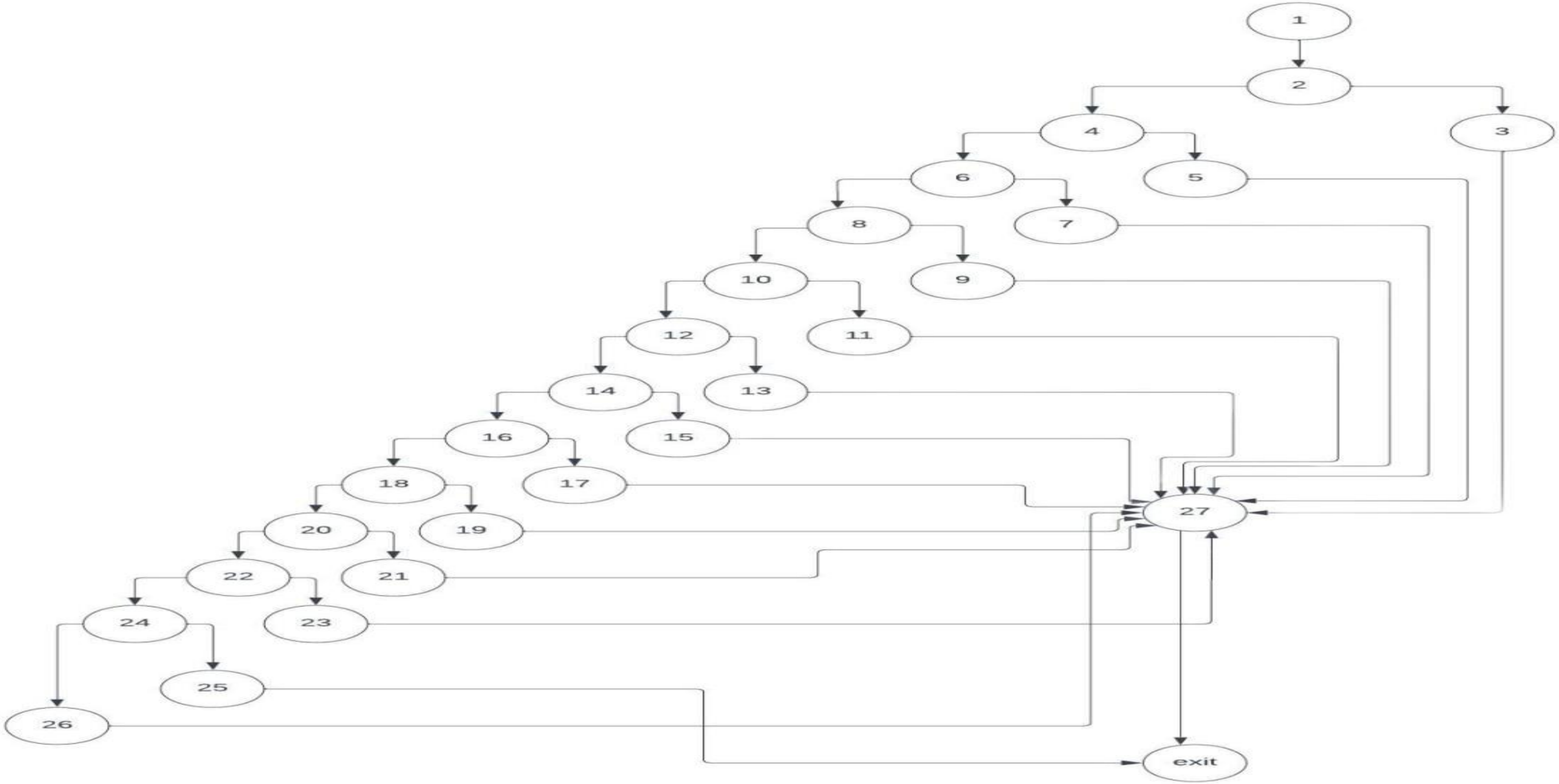


```
12  } else if (totalMark >= 76 && totalMark < 80) {
13      gpa = 2.7;
        grade = "B-";
14  } else if (totalMark >= 73 && totalMark < 76) {
15      gpa = 2.3;
        grade = "C+";
16  } else if (totalMark >= 70 && totalMark < 73) {
17      gpa = 2.0;
        grade = "C";
18  } else if (totalMark >= 67 && totalMark < 70) {
19      gpa = 1.7;
        grade = "C-";
20  } else if (totalMark >= 64 && totalMark < 67) {
21      gpa = 1.3;
        grade = "D+";
22  } else if (totalMark >= 60 && totalMark < 64) {
23      gpa = 1.0;
        grade = "D";
24  } else if (totalMark < 0 || totalMark > 100) {
25      throw new IllegalArgumentException( s: "Invalid total mark");
26  } else {
27      gpa = 0.0;
        grade = "F";
    }

28  return new GPAResult(gpa, grade);
}
```

Testing Type Details: Test GPA Calculations and Grades

Tester Name: Mariam Ahmed

Test Environment Details



Testing Type Details: Test GPA Calculations and Grades
Tester Name: Mariam Ahmed
Test Environment Details

Basis Path Coverage

Path	Test Cases
1 – 2 – 3 – 27 - exit	#1
1 – 2 – 4 - 5 – 27 - exit	#2
1 – 2 – 4 - 6 - 7 – 27 - exit	#3
1 – 2 – 4 - 6 – 8 – 9 – 27 - exit	#4
1 – 2 – 4 – 5 - 6 -8 – 10 - 11 – 27 - exit	#5
1 – 2 – 4 – 5 - 6 -8 – 10 – 12- 13– 27 - exit	#6
1 – 2 – 4 – 5 - 6 -8 – 10 – 12- 14 – 15 – 27 - exit	#7
1 – 2 – 4 – 5 - 6 -8 – 10 – 12- 14 – 16 - 17 – 27 - exit	#8
1 – 2 – 4 – 5 - 6 -8 – 10 – 12- 14 – 16 – 18 – 19 – 27 - exit	#9
1 – 2 – 4 – 5 - 6 -8 – 10 – 12- 14 – 16 – 18 – 20 - 21 – 27 - exit	#10
1 – 2 – 4 – 5 - 6 -8 – 10 – 12- 14 – 16 – 18 – 20 – 22 -23 – 27 - exit	#11
1 – 2 – 4 – 5 - 6 -8 – 10 – 12- 14 – 16 – 18 – 20 – 22 -24 – 25 - exit	#13, #14
1 – 2 – 4 – 5 - 6 -8 – 10 – 12- 14 – 16 – 18 – 20 – 22 -24 – 26 - 27 - exit	#12

N - Number of Nodes = 28
E – No of Edges = 39
V(G) = E – N + 2 = 39 – 28 + 2 = 13
Number of basis Paths = 13

No. of closed Regions = 12
Number of basis Paths = 12 + 1 = 13

Number of basis paths =13
Number of covered basis paths = 13 So basis path coverage 100%
As Basis Path coverage implies Branch coverage So Branch coverage 100% and As Branch coverage implies statement coverage so Statement coverage 100%

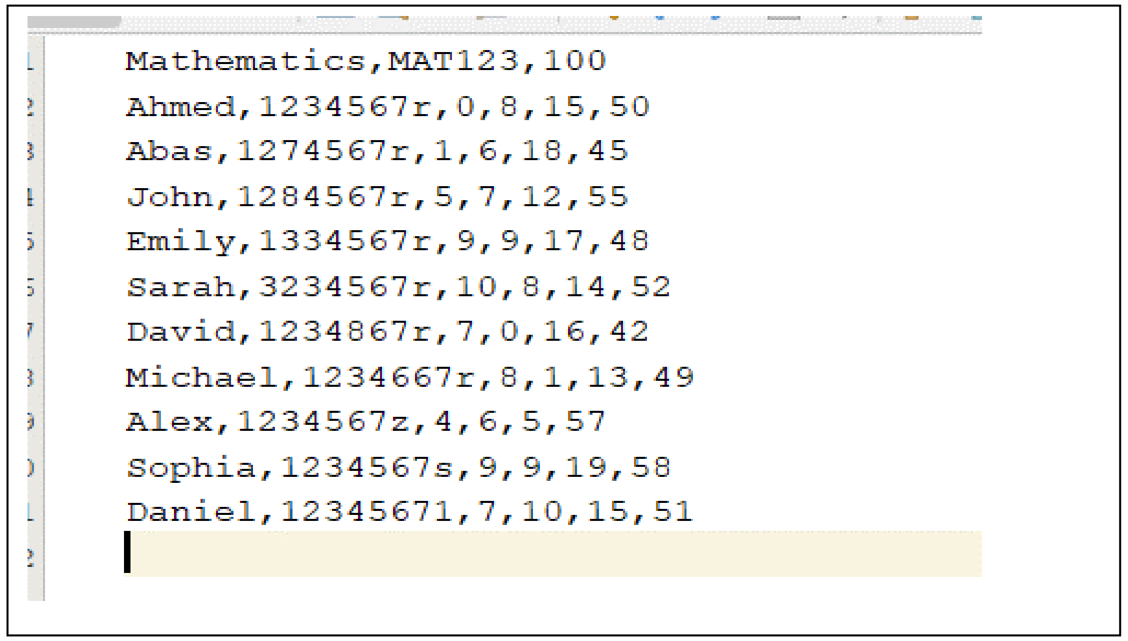
Black Testing

Boundary Testing Technique

Testing Student Activities mark & Oral/Practical mark:

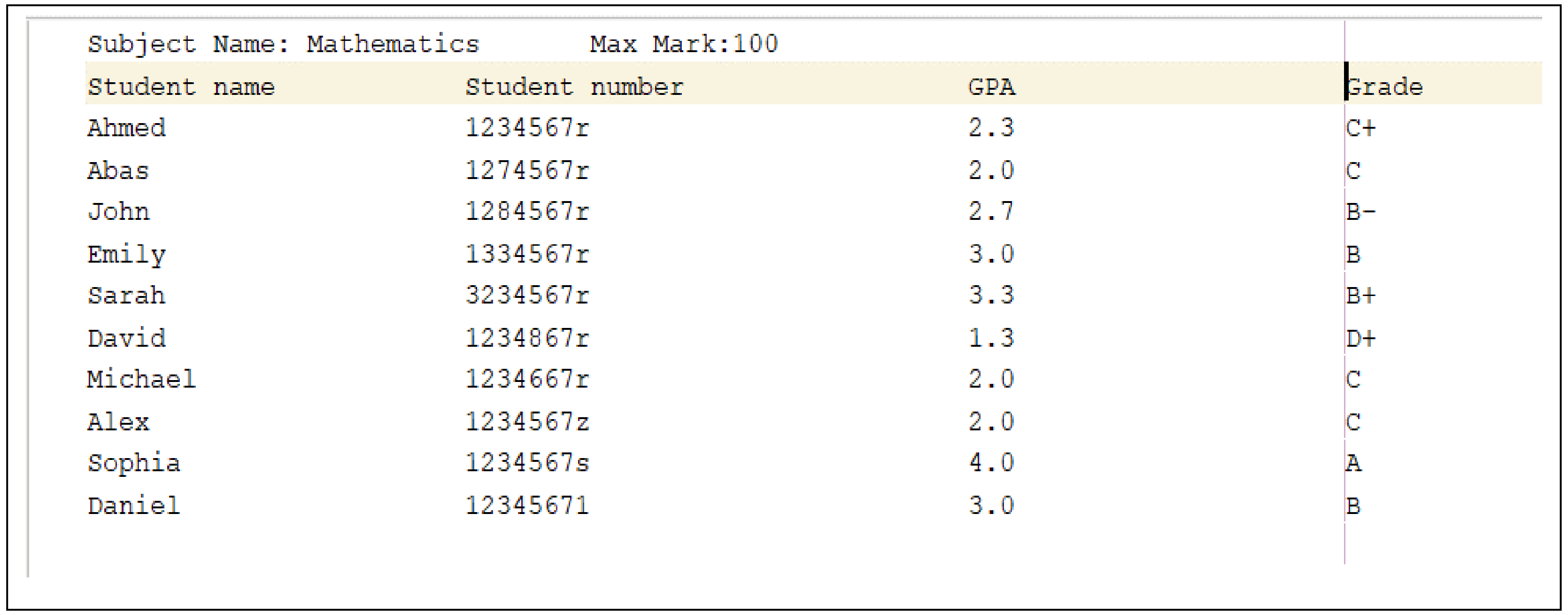
each of them accepts inputs from 0 to 10

input file:



```
Mathematics,MAT123,100
Ahmed,1234567r,0,8,15,50
Abas,1274567r,1,6,18,45
John,1284567r,5,7,12,55
Emily,1334567r,9,9,17,48
Sarah,3234567r,10,8,14,52
David,1234867r,7,0,16,42
Michael,1234667r,8,1,13,49
Alex,1234567z,4,6,5,57
Sophia,1234567s,9,9,19,58
Daniel,12345671,7,10,15,51
```

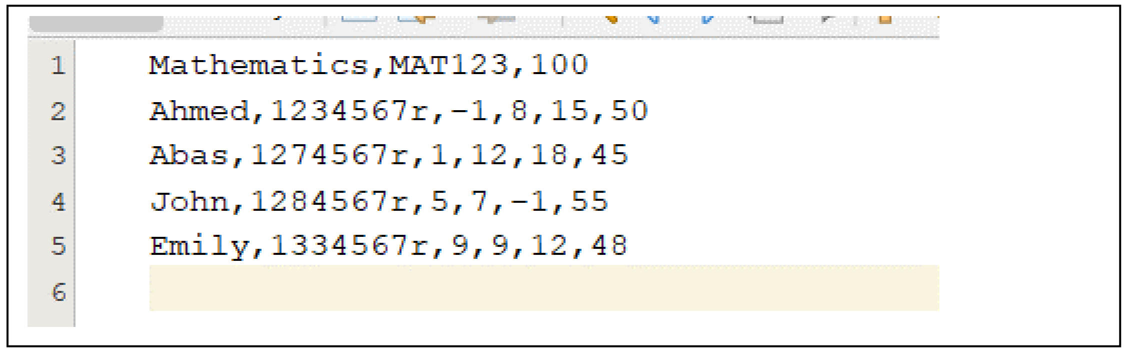
output file :



```
Subject Name: Mathematics       Max Mark:100
Student name            Student number                GPA                     Grade
Ahmed                   1234567r                      2.3                     C+
Abas                    1274567r                      2.0                     C
John                    1284567r                      2.7                     B-
Emily                   1334567r                      3.0                     B
Sarah                   3234567r                      3.3                     B+
David                   1234867r                      1.3                     D+
Michael                 1234667r                      2.0                     C
Alex                    1234567z                      2.0                     C
Sophia                  1234567s                      4.0                     A
Daniel                  12345671                      3.0                     B
```

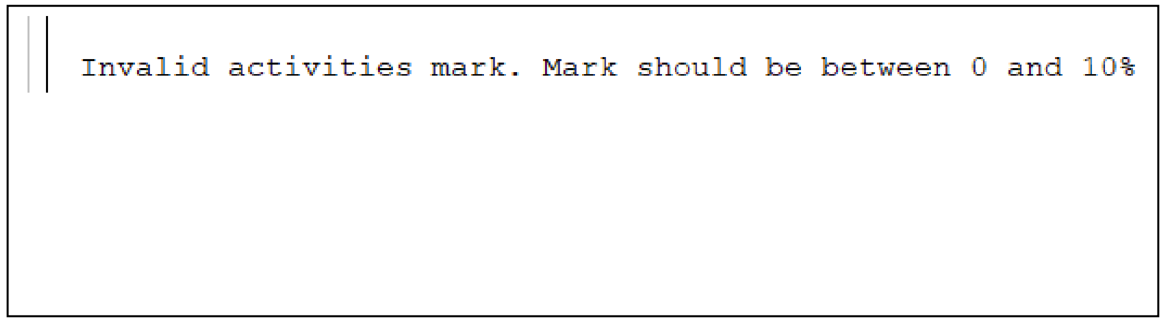
When test invalid input like -1 or 12 it isn't generated the output file

input file:



```
Mathematics,MAT123,100
Ahmed,1234567r,-1,8,15,50
Abas,1274567r,1,12,18,45
John,1284567r,5,7,-1,55
Emily,1334567r,9,9,12,48
```

output file :

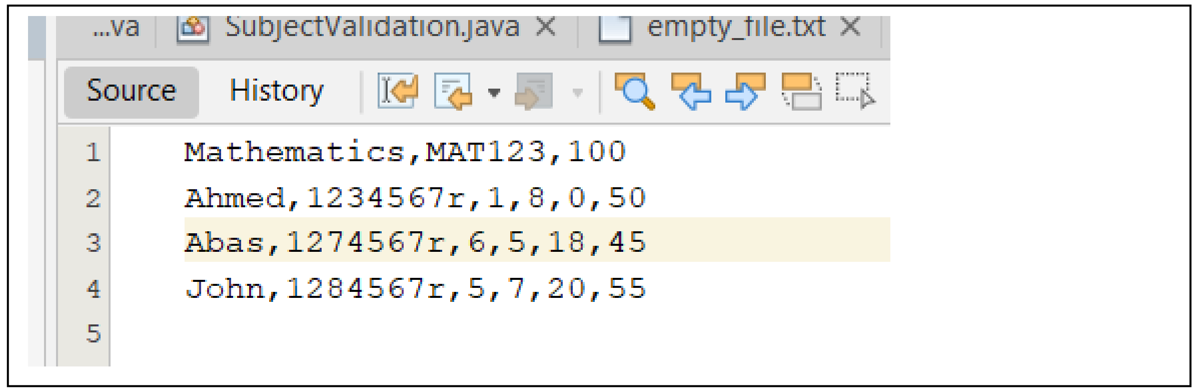


```
Invalid activities mark. Mark should be between 0 and 10%
```

Testing Student Midterm exam mark:

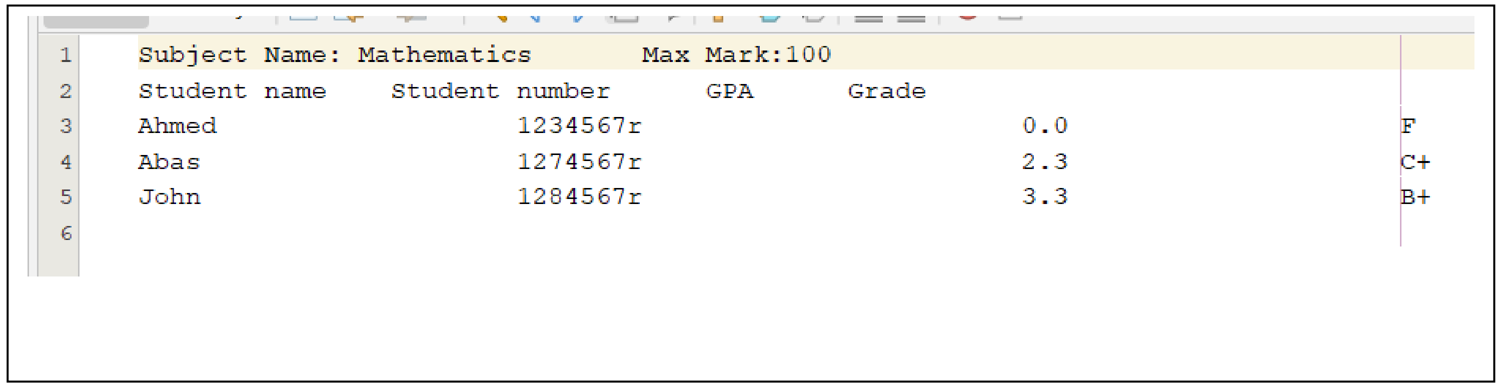
It takes values from 0 to 20

input file:



```
Mathematics,MAT123,100
Ahmed,1234567r,1,8,0,50
Abas,1274567r,6,5,18,45
John,1284567r,5,7,20,55
```

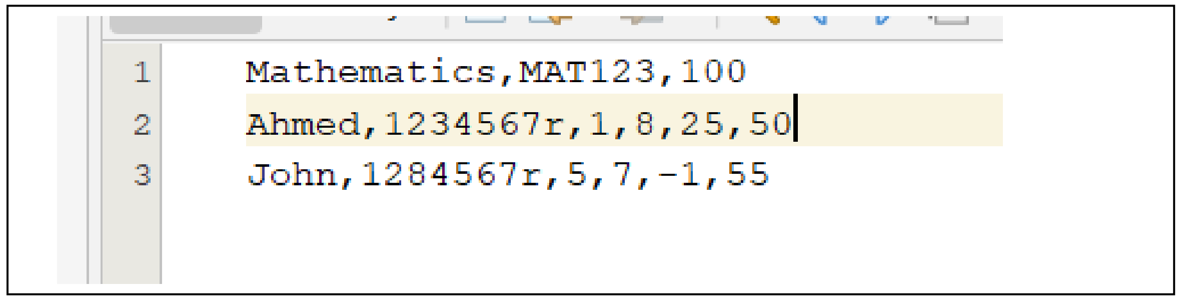
output file:



```
Subject Name: Mathematics          Max Mark:100
Student name    Student number     GPA      Grade
Ahmed                   1234567r                     0.0                        F
Abas                    1274567r                     2.3                        C+
John                    1284567r                     3.3                        B+
```

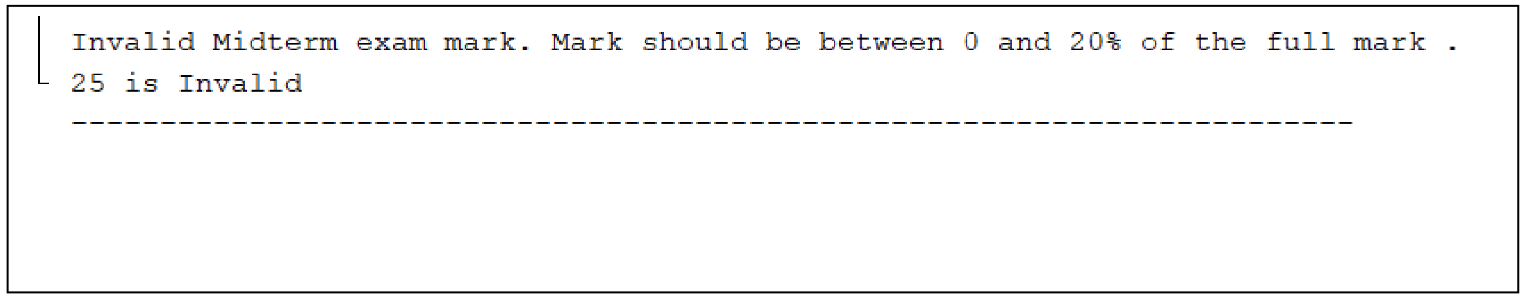
When test invalid input like -1 or 25 it isn't generated the output file

input file:



```
Mathematics,MAT123,100
Ahmed,1234567r,1,8,25,50
John,1284567r,5,7,-1,55
```

output:



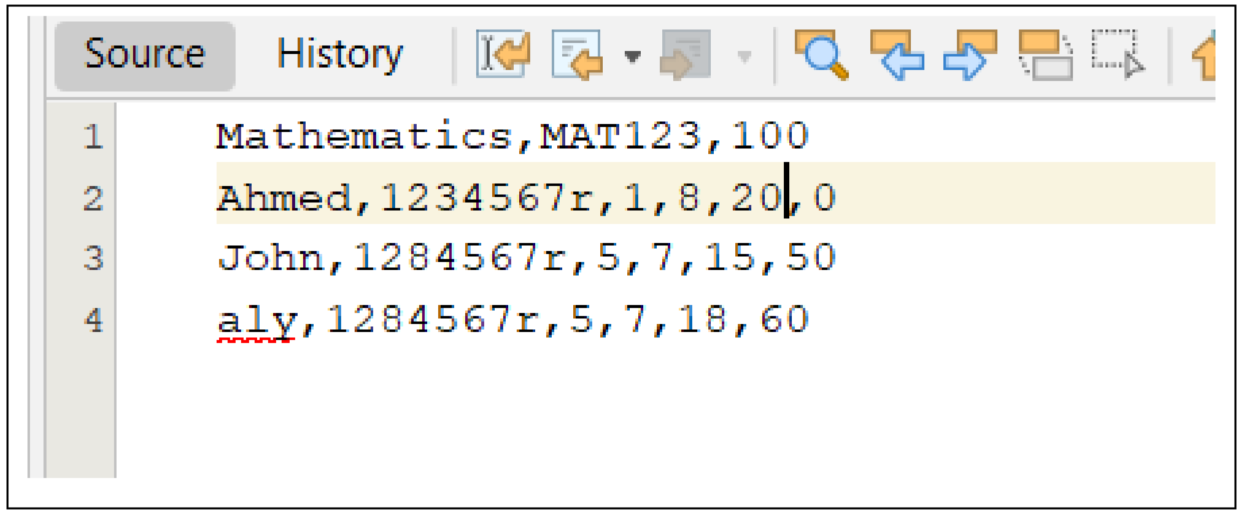
```
Invalid Midterm exam mark. Mark should be between 0 and 20% of the full mark .
25 is Invalid
-------------------------------------------------------------------------
```

Testing Student Final exam mark:

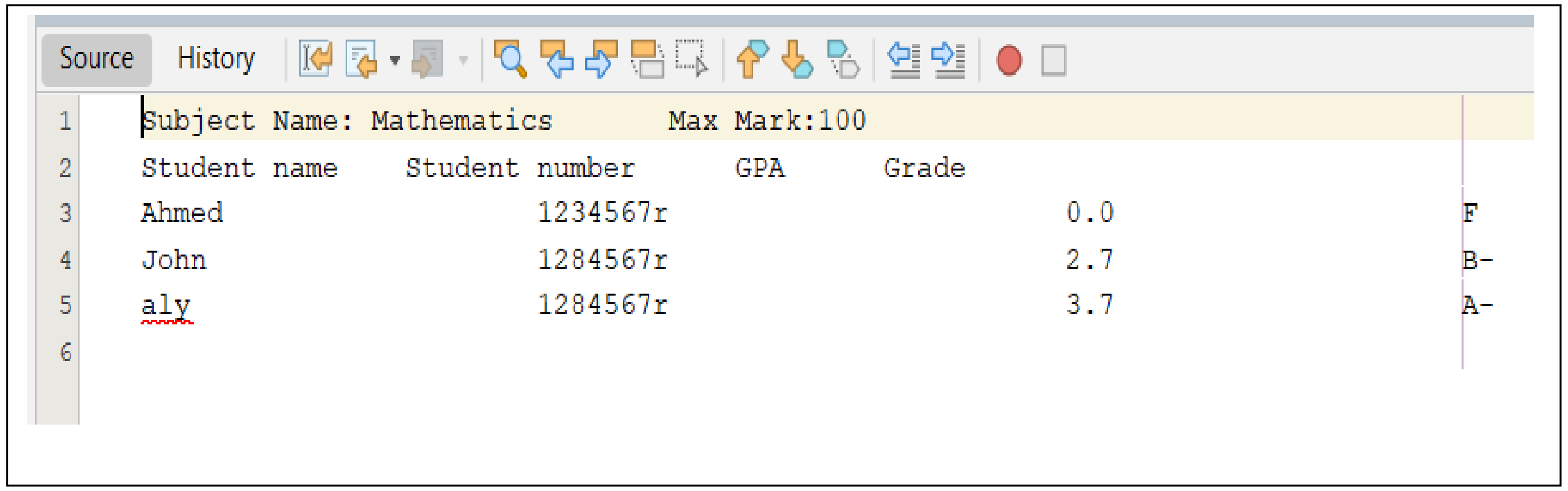
It takes values from 0 to 60

input file:

output file:



```
Mathematics,MAT123,100
Ahmed,1234567r,1,8,20,0
John,1284567r,5,7,15,50
aly,1284567r,5,7,18,60
```

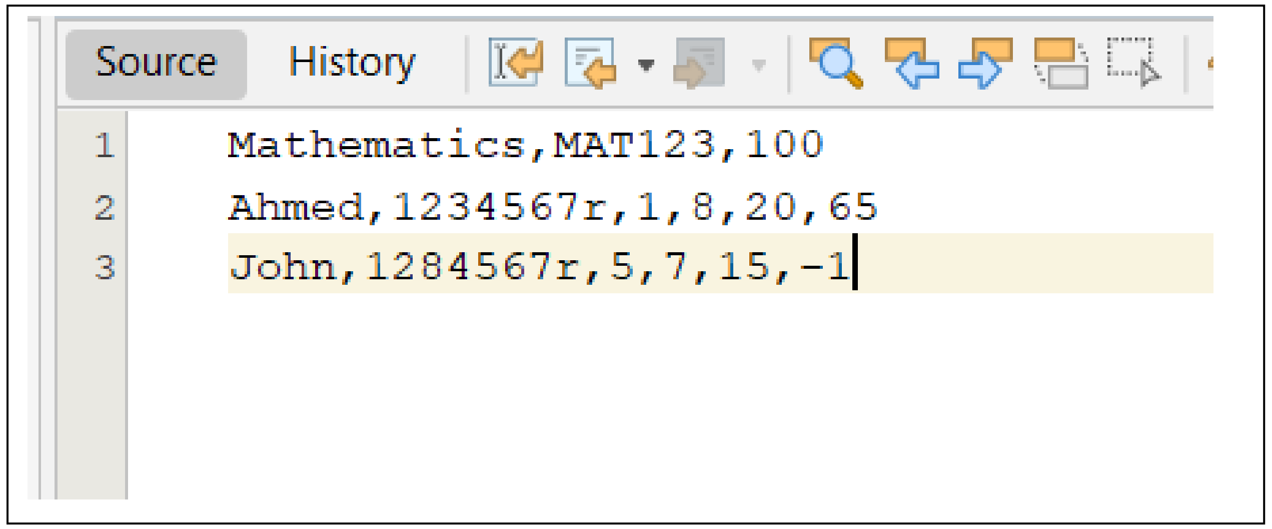


```
Subject Name: Mathematics        Max Mark:100
Student name     Student number       GPA       Grade
Ahmed                     1234567r                    0.0                            F
John                      1284567r                    2.7                            B-
aly                       1284567r                    3.7                            A-
```

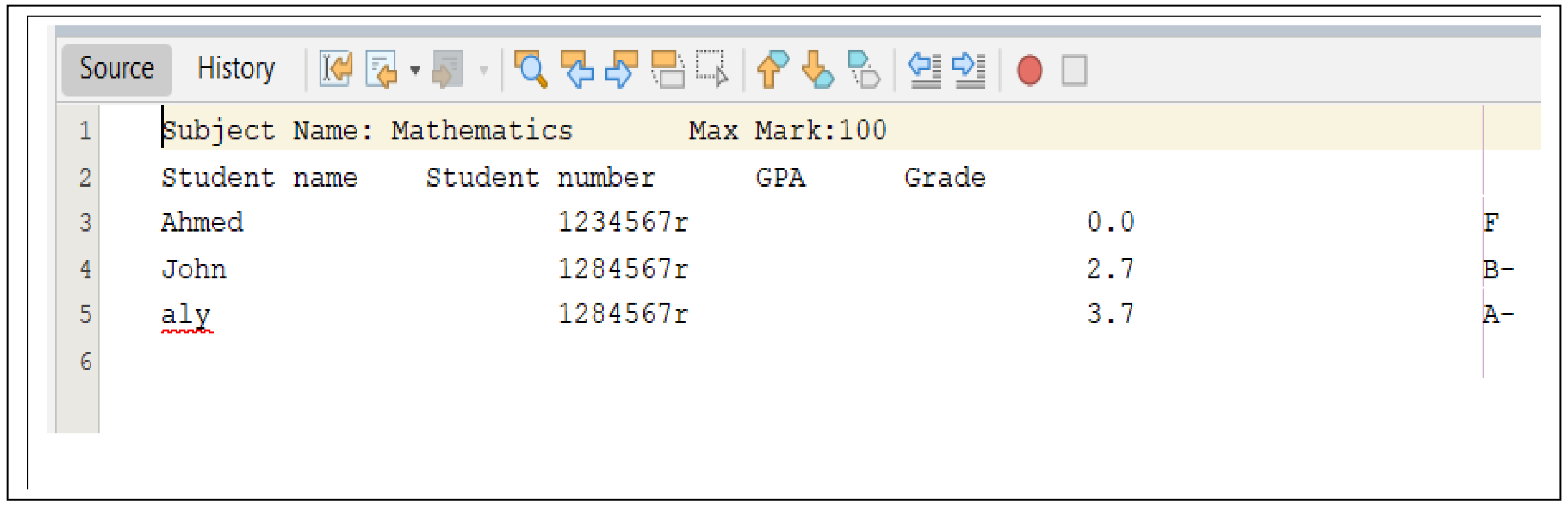
When test invalid input like -1 or 60 it isn’t generated the output file:

input file:

output:



```
Mathematics,MAT123,100
Ahmed,1234567r,1,8,20,65
John,1284567r,5,7,15,-1
```



```
Subject Name: Mathematics        Max Mark:100
Student name     Student number       GPA       Grade
Ahmed                     1234567r                    0.0                            F
John                      1284567r                    2.7                            B-
aly                       1284567r                    3.7                            A-
```

Testing Type Details: Equivalence class testing
Tester Name: Sara Ashraf
Test Environment Details

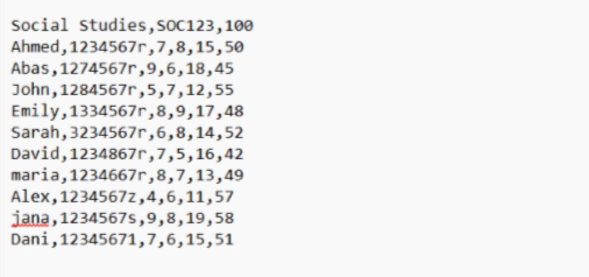
Equivalence Class Partitioning

1-Full mark

- Valid classes:

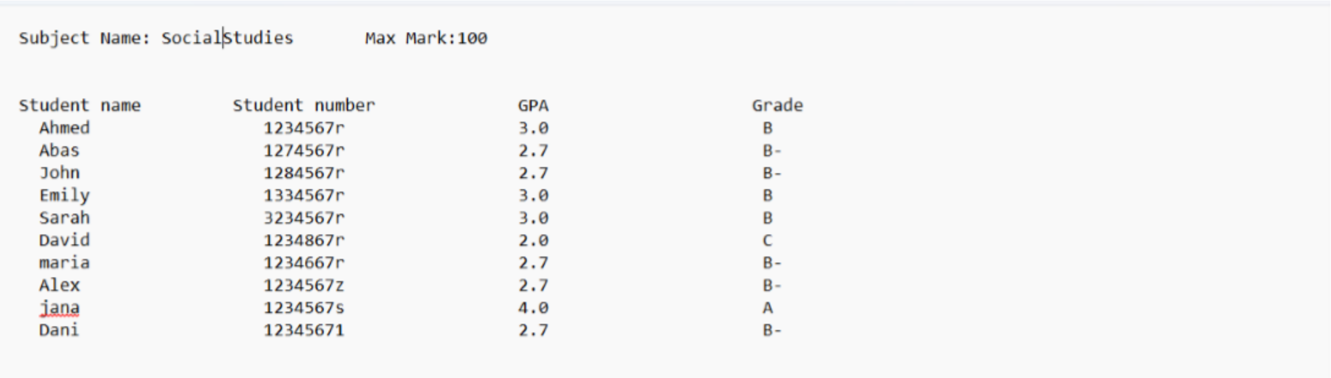
class 1: a numeric value of 100::.

Input



```
Social Studies,SOC123,100
Ahmed,1234567r,7,8,15,50
Abas,1274567r,9,6,18,45
John,1284567r,5,7,12,55
Emily,1334567r,8,9,17,48
Sarah,3234567r,6,8,14,52
David,1234867r,7,5,16,42
maria,1234667r,8,7,13,49
Alex,1234567z,4,6,11,57
jana,1234567s,9,8,19,58
Dani,12345671,7,6,15,51
```

output



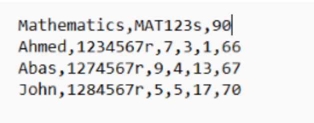
```
Subject Name: SocialStudies       Max Mark:100


Student name         Student number            GPA                    Grade
  Ahmed                 1234567r               3.0                     B
  Abas                  1274567r               2.7                     B-
  John                  1284567r               2.7                     B-
  Emily                 1334567r               3.0                     B
  Sarah                 3234567r               3.0                     B
  David                 1234867r               2.0                     C
  maria                 1234667r               2.7                     B-
  Alex                  1234567z               2.7                     B-
  jana                  1234567s               4.0                     A
  Dani                  12345671               2.7                     B-
```

Testing Type Details: Equivalence class testing
Tester Name: Sara Ashraf
Test Environment Details

- Invalid Classes

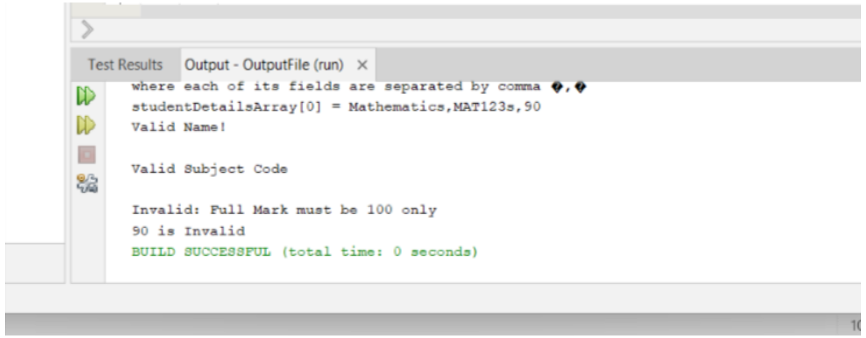
class 1: a numeric value of less than 100

Input



```
Mathematics,MAT123s,90
Ahmed,1234567r,7,3,1,66
Abas,1274567r,9,4,13,67
John,1284567r,5,5,17,70
```

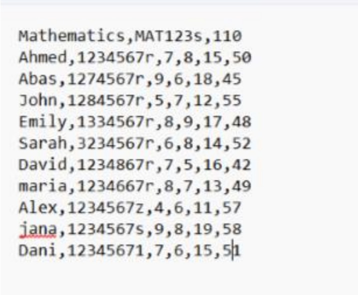
output



```
where each of its fields are separated by comma �,�
studentDetailsArray[0] = Mathematics,MAT123s,90
Valid Name!

Valid Subject Code

Invalid: Full Mark must be 100 only
90 is Invalid
BUILD SUCCESSFUL (total time: 0 seconds)
```

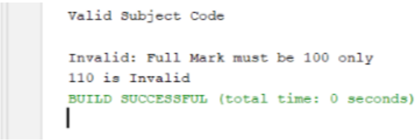
class 2: a numeric value of morethan 100

Input



```
Mathematics,MAT123s,110
Ahmed,1234567r,7,8,15,50
Abas,1274567r,9,6,18,45
John,1284567r,5,7,12,55
Emily,1334567r,8,9,17,48
Sarah,3234567r,6,8,14,52
David,1234867r,7,5,16,42
maria,1234667r,8,7,13,49
Alex,1234567z,4,6,11,57
jana,1234567s,9,8,19,58
Dani,12345671,7,6,15,51
```

output



```
Valid Subject Code

Invalid: Full Mark must be 100 only
110 is Invalid
BUILD SUCCESSFUL (total time: 0 seconds)
```

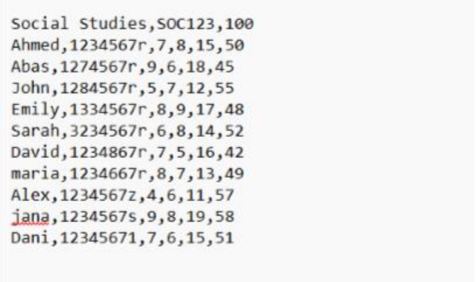
Testing Type Details: Equivalence class testing
Tester Name: Sara Ashraf
Test Environment Details

2- Student Activities marks::

- Valid classes:

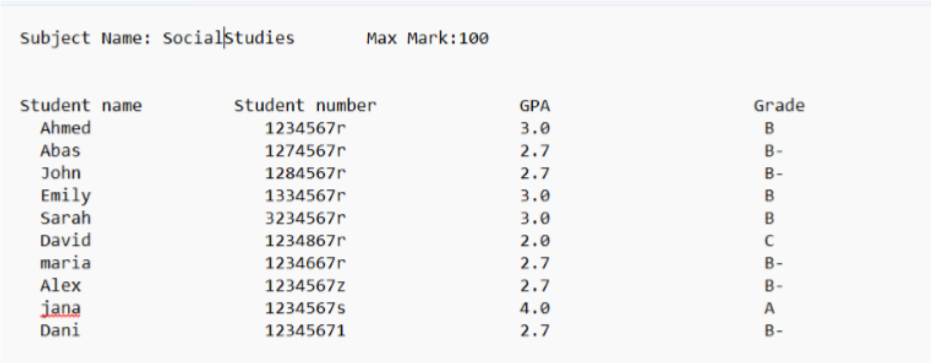
class 1: integer between 0 and 10.

Input



```
Social Studies,SOC123,100
Ahmed,1234567r,7,8,15,50
Abas,1274567r,9,6,18,45
John,1284567r,5,7,12,55
Emily,1334567r,8,9,17,48
Sarah,3234567r,6,8,14,52
David,1234867r,7,5,16,42
maria,1234667r,8,7,13,49
Alex,1234567z,4,6,11,57
jana,1234567s,9,8,19,58
Dani,12345671,7,6,15,51
```

output



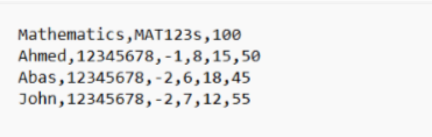
```
Subject Name: SocialStudies        Max Mark:100


Student name          Student number          GPA               Grade
  Ahmed                  1234567r              3.0                B
  Abas                   1274567r              2.7                B-
  John                   1284567r              2.7                B-
  Emily                  1334567r              3.0                B
  Sarah                  3234567r              3.0                B
  David                  1234867r              2.0                C
  maria                  1234667r              2.7                B-
  Alex                   1234567z              2.7                B-
  jana                   1234567s              4.0                A
  Dani                   12345671              2.7                B-
```

- Invalid Classes

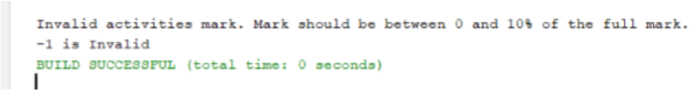
class 1: integer less than 0.

Input

output

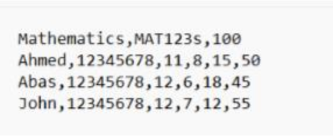


```
Mathematics,MAT123s,100
Ahmed,12345678,-1,8,15,50
Abas,12345678,-2,6,18,45
John,12345678,-2,7,12,55
```

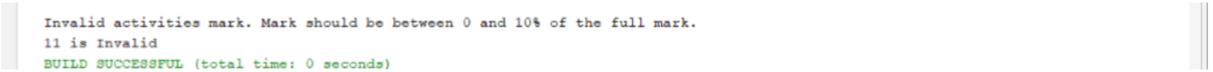


```
Invalid activities mark. Mark should be between 0 and 10% of the full mark.
-1 is Invalid
BUILD SUCCESSFUL (total time: 0 seconds)
```

class 2: integer more than 10.



```
Mathematics,MAT123s,100
Ahmed,12345678,11,8,15,50
Abas,12345678,12,6,18,45
John,12345678,12,7,12,55
```

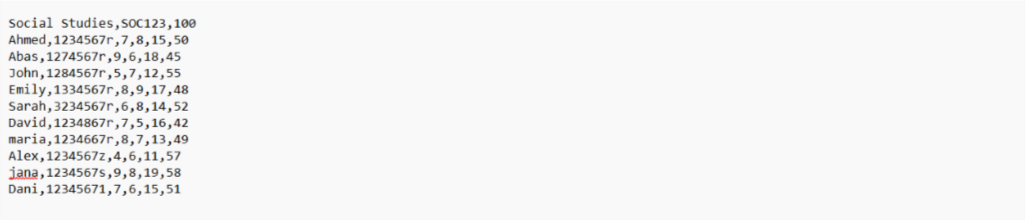


```
Invalid activities mark. Mark should be between 0 and 10% of the full mark.
11 is Invalid
BUILD SUCCESSFUL (total time: 0 seconds)
```

7- Oral/Practical mark:

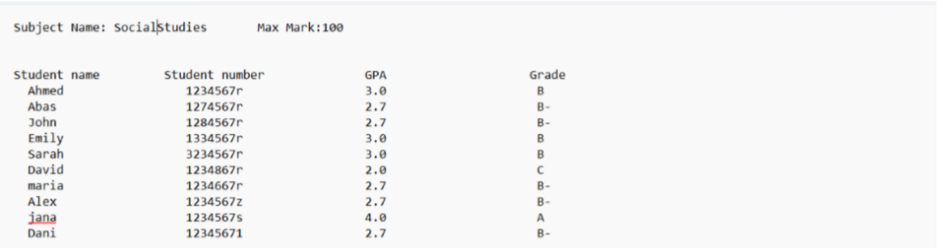
- Valid classes:
- class 1: integer between 0 and 10.

Input Screenchot:



```
Social Studies,SOC123,100
Ahmed,1234567r,7,8,15,50
Abas,1274567r,9,6,18,45
John,1284567r,5,7,12,55
Emily,1334567r,8,9,17,48
Sarah,3234567r,6,8,14,52
David,1234867r,7,5,16,42
maria,1234667r,8,7,13,49
Alex,1234567z,4,6,11,57
jana,1234567s,9,8,19,58
Dani,12345671,7,6,15,51
```

Output Screenchot:

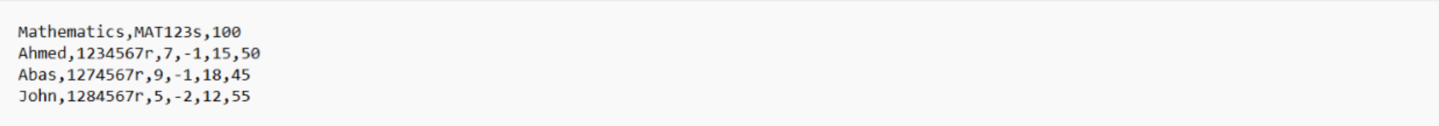


```
Subject Name: SocialStudies       Max Mark:100


Student name        Student number             GPA                 Grade
  Ahmed                1234567r                3.0                  B
  Abas                 1274567r                2.7                  B-
  John                 1284567r                2.7                  B-
  Emily                1334567r                3.0                  B
  Sarah                3234567r                3.0                  B
  David                1234867r                2.0                  C
  maria                1234667r                2.7                  B-
  Alex                 1234567z                2.7                  B-
  jana                 1234567s                4.0                  A
  Dani                 12345671                2.7                  B-
```

Invalid Classes

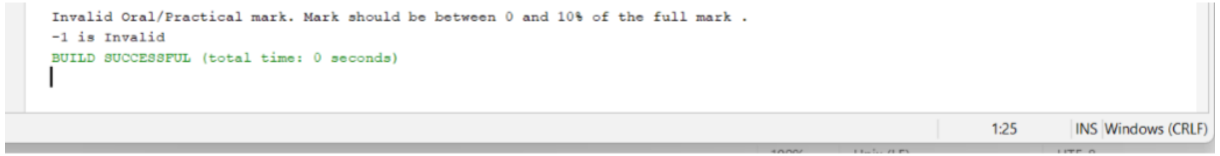
- class 1: integer less than 0.

Input Screenchot:



```
Mathematics,MAT123s,100
Ahmed,1234567r,7,-1,15,50
Abas,1274567r,9,-1,18,45
John,1284567r,5,-2,12,55
```

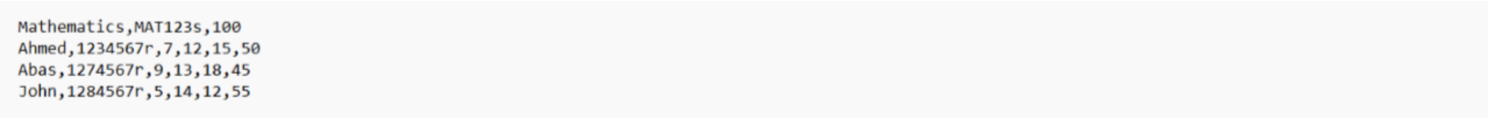
Output Screenchot:



```
Invalid Oral/Practical mark. Mark should be between 0 and 10% of the full mark .
-1 is Invalid
BUILD SUCCESSFUL (total time: 0 seconds)
```

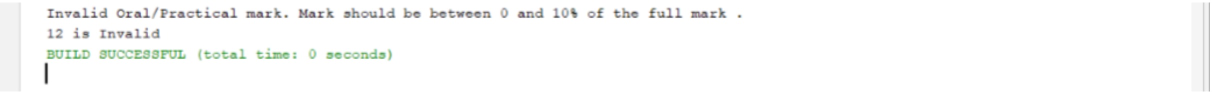
class 2: Alphabetic characters and Spaces start with space.

Input Screenchot:



```
Mathematics,MAT123s,100
Ahmed,1234567r,7,12,15,50
Abas,1274567r,9,13,18,45
John,1284567r,5,14,12,55
```

Output Screenchot:



```
Invalid Oral/Practical mark. Mark should be between 0 and 10% of the full mark .
12 is Invalid
BUILD SUCCESSFUL (total time: 0 seconds)
```

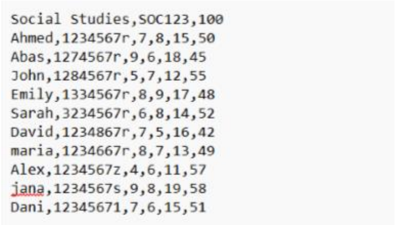
8-Midterm exam mark:

- Valid classes:

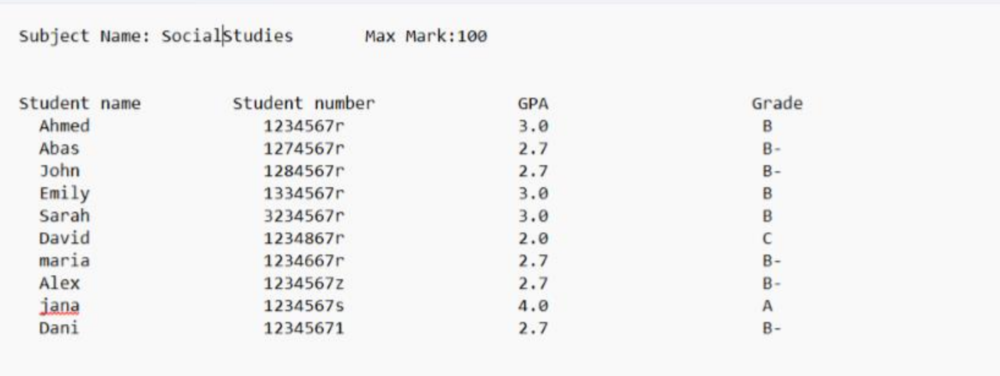
class 1: integer between 0 and 20.

Input

output



```
Social Studies,SOC123,100
Ahmed,1234567r,7,8,15,50
Abas,1274567r,9,6,18,45
John,1284567r,5,7,12,55
Emily,1334567r,8,9,17,48
Sarah,3234567r,6,8,14,52
David,1234867r,7,5,16,42
maria,1234667r,8,7,13,49
Alex,1234567z,4,6,11,57
jana,1234567s,9,8,19,58
Dani,12345671,7,6,15,51
```

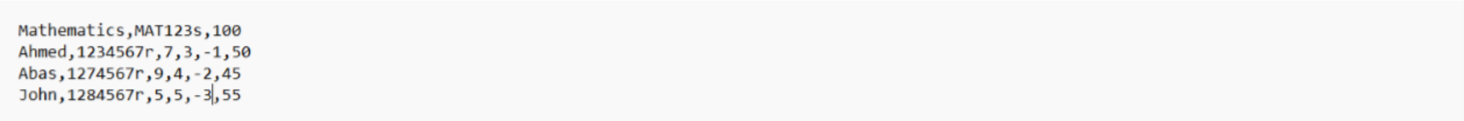


```
Subject Name: SocialStudies      Max Mark:100


Student name        Student number            GPA                Grade
  Ahmed                1234567r               3.0                  B
  Abas                 1274567r               2.7                  B-
  John                 1284567r               2.7                  B-
  Emily                1334567r               3.0                  B
  Sarah                3234567r               3.0                  B
  David                1234867r               2.0                  C
  maria                1234667r               2.7                  B-
  Alex                 1234567z               2.7                  B-
  jana                 1234567s               4.0                  A
  Dani                 12345671               2.7                  B-
```

- Invalid Classes

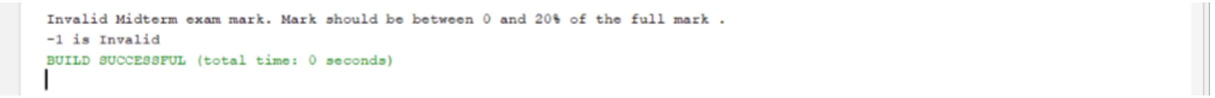
class 1: integer less than 0.

Input Screenchot:



```
Mathematics,MAT123s,100
Ahmed,1234567r,7,3,-1,50
Abas,1274567r,9,4,-2,45
John,1284567r,5,5,-3,55
```

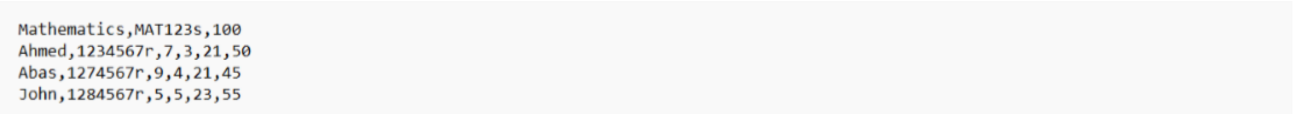
Output Screenchot:



```
Invalid Midterm exam mark. Mark should be between 0 and 20% of the full mark .
-1 is Invalid
BUILD SUCCESSFUL (total time: 0 seconds)
```

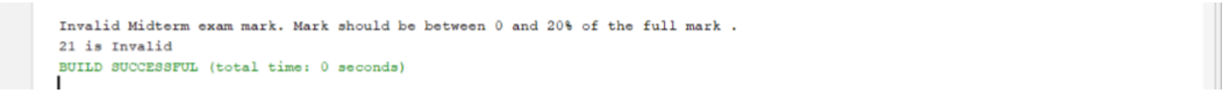
class 2: Alphabetic characters and Spaces start with space.

Input Screenchot:



```
Mathematics,MAT123s,100
Ahmed,1234567r,7,3,21,50
Abas,1274567r,9,4,21,45
John,1284567r,5,5,23,55
```

Output Screenchot:



```
Invalid Midterm exam mark. Mark should be between 0 and 20% of the full mark .
21 is Invalid
BUILD SUCCESSFUL (total time: 0 seconds)
```

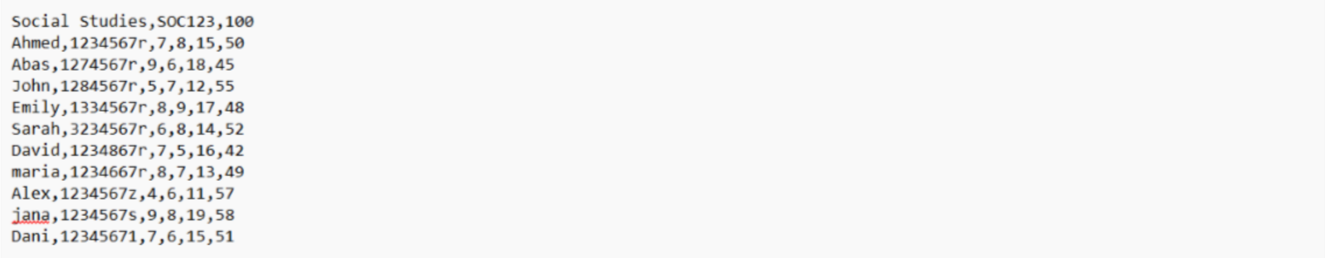
Final exam mark:

- Valid classes:

Testing Type Details: Equivalence class testing
Tester Name: Sara Ashraf
Test Environment Details

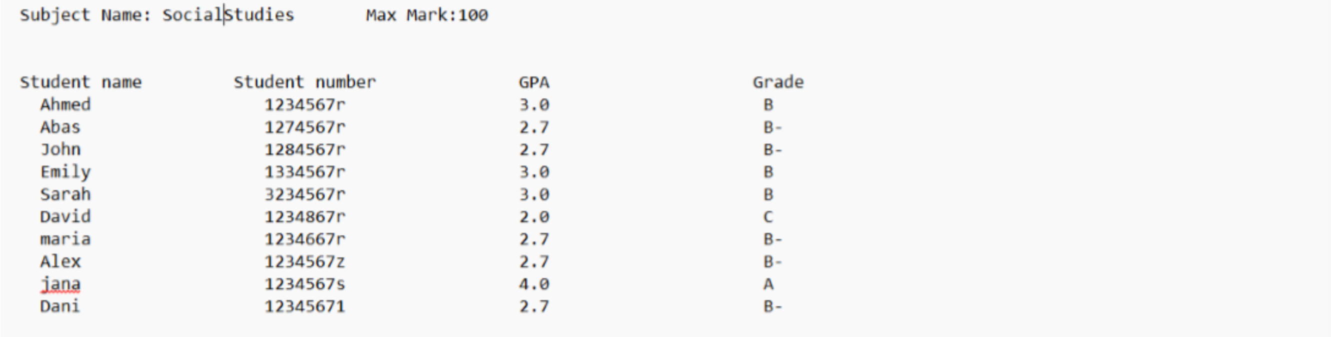
class 1: integer between 0 and 60:.

Input Screenchot:



```
Social Studies,SOC123,100
Ahmed,1234567r,7,8,15,50
Abas,1274567r,9,6,18,45
John,1284567r,5,7,12,55
Emily,1334567r,8,9,17,48
Sarah,3234567r,6,8,14,52
David,1234867r,7,5,16,42
maria,1234667r,8,7,13,49
Alex,1234567z,4,6,11,57
jana,1234567s,9,8,19,58
Dani,12345671,7,6,15,51
```

Output Screenchot:

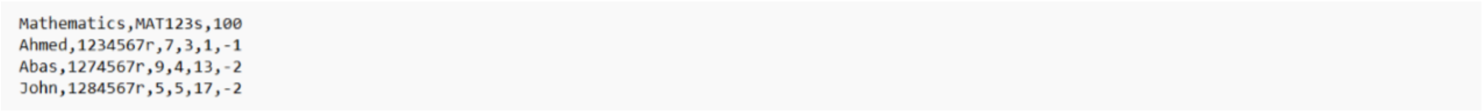


```
Subject Name: SocialStudies       Max Mark:100


Student name        Student number            GPA                     Grade
  Ahmed                1234567r               3.0                      B
  Abas                 1274567r               2.7                      B-
  John                 1284567r               2.7                      B-
  Emily                1334567r               3.0                      B
  Sarah                3234567r               3.0                      B
  David                1234867r               2.0                      C
  maria                1234667r               2.7                      B-
  Alex                 1234567z               2.7                      B-
  jana                 1234567s               4.0                      A
  Dani                 12345671               2.7                      B-
```

- Invalid Classes

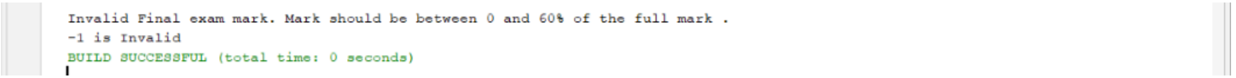
class 1: integer less than 0.

Input Screenchot:



```
Mathematics,MAT123s,100
Ahmed,1234567r,7,3,1,-1
Abas,1274567r,9,4,13,-2
John,1284567r,5,5,17,-2
```

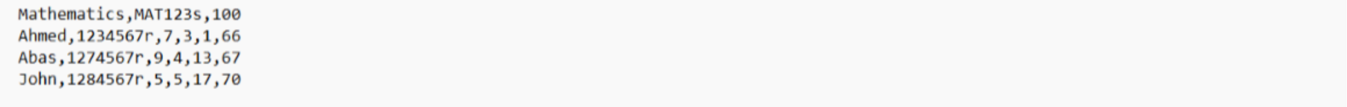
Output Screenchot:



```
Invalid Final exam mark. Mark should be between 0 and 60% of the full mark .
-1 is Invalid
BUILD SUCCESSFUL (total time: 0 seconds)
```

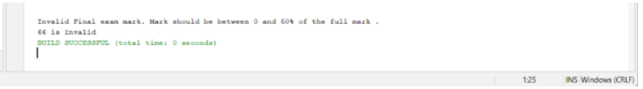
class 2: Alphabetic characters and Spaces start with space.

Input Screenchot:



```
Mathematics,MAT123s,100
Ahmed,1234567r,7,3,1,66
Abas,1274567r,9,4,13,67
John,1284567r,5,5,17,70
```

Output Screenchot:



```
Invalid Final exam mark. Mark should be between 0 and 60% of the full mark .
66 is Invalid
BUILD SUCCESSFUL (total time: 0 seconds)
```