# Computing testing

## Testing tables

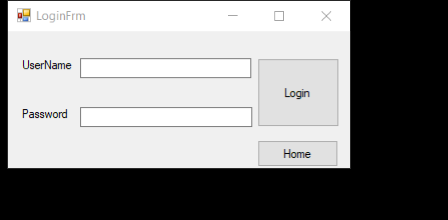
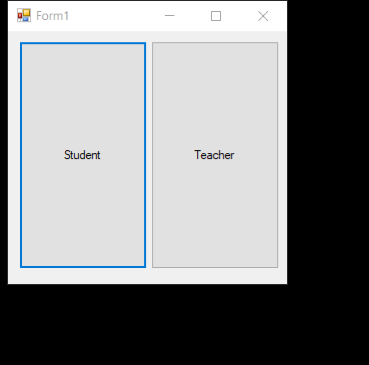
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test | Purpose | Description | Data | Expected result | Actual result | Pass/Fail |
| Form1 | | | | | | |
| 1 | Navigation | Navigate to Login form | Teacher button click | Login form displayed | Login form displayed | Pass |
| 2 | Navigation | Navigate to student form | Student button click | Student form displayed | Student form displayed | pass |
| Login Form | | | | | | |
| 1 | Log into teacher side | Incorrect username correct password | Username: Testdata [invalid]  Password: Pass | Message box with telling user incorrect username or password | Message box telling user incorrect username or password | Pass |
| 2 | Log into teacher side | Correct Username incorrect password | Username: TTest1  Password: TestData [invalid] | Message box telling user incorrect username or password | Message box telling user incorrect username or password | pass |
| 3 | Log in to teacher side | Correct Username and password | Username: TTest1  Password: Pass | Display teacher form with students filled into text boxes filtered by the teacher’s class and add student enabled | Displays Teacher form with students filled into text boxes filtered by the teacher’s class and add student enabled | pass |
| 4 | Log into teacher side with Admin account | Attempting to log in with admin account | Username: Admin  Password: Admin | Display teacher form with all students and add teacher enabled. | Display teacher form with all students and add teacher enabled. | pass |
| 5 | Login missing Username | Attempting to login without all fields | Username: <blank>  Password:  pass | Displays message box prompting to fill missing fields | Displays message box prompting to fill missing fields | pass |
| 6 | Login missing password | Attempting to login without all fields | Username:  TTest1  Password:  <blank> | Displays message box prompting to fill missing fields | Displays message box prompting to fill missing fields | Pass |
| 7 | Sql injection | Test if parameterised sql works | Username = “or””=”  Password –  “or”” =”  Taken from - https://www .w3schools. com/sql/sql injection.asp | Message box telling user incorrect username or password, doesn’t log in | Message box telling user incorrect username or password, doesn’t log in | pass |
| Teacher Form | | | | | | |
| 1 | Changes multiple score being displayed | Changes score displayed | Combo box select 3 | Displayed score for the multiple of 3 of the students (10) from multiple of 2 (0) | Displayed score for the multiple of 3 of the students (10) from multiple of 2 (0) | pass |
| 2 | Filters students into correct class | Filters students into class of logged in teacher | Login in with teacher of class one.  (Login form)  Username = TTest1  Password = Pass | Only displays students of class 1 | Only displays students of class one | pass |
| 3 | Changes students record in class in ascending order of StudentID | Students are all displayed from the teachers class | Button click increase, from Student record 1 | Display studentId record 2 | Displays StudentId record 2 | pass |
| 4 | Change Student record | Decrease Student record by Id | Button click Decrease, from Student record 2 | Display studentId record 1 | Displays StudentId record 1 | pass |
| 5 | Remove Record | Correct removal of current record | StudentId = 3 selected, remove student clicked | Message box display asking of they would like to continue, once yes click form reloaded not evidence of record removed | Message box display asking of they would like to continue, once yes click form reloaded not evidence of record removed | pass |
| 6 | Average | Correct average score calculated | All student scores, Selected multiple – 2 Currently  StudentId -1  Score - 75  StudentId -2  Score - 0  StudentId -4  Score -0 | Average displays - 25  Total score/number of students | Average score displayed = 25 | pass |
| Add Student | | | | | | |
| 1 | Add student missing fields | Checks if all need information is present | First name = <Blank>  Surname = <Blank>  Password =  <Blank> | Message box displayed prompting to fill all fields | Message box displayed prompting to fill all fields | pass |
| 2 | Adding Student with same first and last name, of existing student | Unique Usernames | 1)  First name = Test  Surname = Test  2)  First name = Test  Surname = Test | Student 1 created username = TTest1  Student 2 Created username = TTest2 | Student ones created username = TTest1  Student twos Created username = TTest2 | pass |
| 3 | Adding student | Student added with all correct details | 1)  First name = Finn  Surname = Rea  Password = Pass | New student added to teachers’ class. Username = FRea1  Password = Pass | New student added to teachers’ class. Username = FRea1  Password = Pass | pass |
| 4 | Hashing | Password hashed | New student  Username – Test  Surname – Test  Password – Cross1 | Password is hashed before stored in data table StudentUsers | Password is hashed before stored in data table StudentUsers | pass |
| Teacher add (From admin account) | | | | | | |
| 1 | Adding teacher missing fields | Checks if all information required to add teacher is present | First name = <Blank>  Surname = <Blank>  Password =  <Blank>  Class = <blank> | Message box displayed prompting to fill all fields | Message box displayed prompting to fill all fields | pass |
| 2 | Adding Teacher with same first and last name of pre-existing teacher | Unique Username | (1)  First name = Test  Surname = Test  (2)  First name = Test  Surname = Test | Teacher 1 created username = TTTest1  Teacher 2 Created username = TTTest2 | Teacher 1 created username = TTTest1  Teacher 2 Created username = TTTest2 | pass |
| 3 | Teacher add with all valid fields | New Teacher | First name = Finn  Surname = Rea  Password =  pass  Class = 3 | New Teacher add with Username =TFRea1  Password =Pass  Class =1 | New Teacher add with Username =TFRea1  Password =Pass  Class =1 | pass |
| 4 | Hashing | Password hashed on creation | New teacher  Username – Test  Password – Test | New teacher and hashed password stored in datatable Users | New teacher and hashed password stored in datatable Users | pass |
| VeiwDisplayForm | | | | | | |
| 1 | Display | Grid displays all student scores in class | Login as Teacher with class one | Displays all students | Displays all students | pass |
| StudentFrm | | | | | | |
| 1 | Missing login information | Testing Login | Username =  <blank>  Password = Pass | Message box shown “Please fill in username” takes back to login | Message box shown “Please fill in Username” takes back to login | Pass |
| 2 | Incorrect Password | Login for Students with incorrect information | Username = TTest1  Password = Test (incorrect) | Message box shown “Incorrect Login” takes back to login | Message box shown “Incorrect Login” takes back to login | Pass |
| 3 | Incorrect username | Login for students with incorrect information | Username = Incorrect (Incorrect)  Password =Pass | Message box shown “Incorrect Login” | Message box shown “Incorrect Login” | Pass |
| 4 | Correct Login | Logins in correctly | Username = TTest1  Password = Pass | Takes to XNA game | Takes to XNA game | pass |
| 5 | File writing | Filters correct student user and writes correct information about student to text file Student  Information:  First name  Surname  Score2  Score3  Score4  Score5  Score6  Score7  Score8  Score9 | Username = TTest1  Password = Pass2 | Text file (Student) to contain  “  Test  Test  0  0  0  0  0  0  0  0” | Text file (Student) to contain  “  Test  Test  0  0  0  0  0  0  0  0” | pass |
| 6 | Sql injection | Testing parameterised Sql | Username – “or””=”  Password – “or””=” | Message box displayed showing incorrect login | Message box displayed showing incorrect login | pass |
| Game | | | | | | |
| 1 | Read Text file | Reads information from (student.txt) | Game opens | Displays logged in student’s information (TTest1)  First name: Test  Surname: Test  Score2: 70  Score3: 0  Score4: 0  Score5: 0  Score6: 0  Score7: 0  Score8: 0  Score9: 0 | Displays logged in student’s information (TTest1)  First name: Test  Surname: Test  Score2:70  Score3: 0  Score4: 0  Score5: 0  Score6: 0  Score7: 0  Score8: 0  Score9: 0 | Pass  Test Video – 0:20 |
| 2 | Multiple select change | Button increase clicked and doesn’t increase past 9 | Right button clicked 10 times | Multiple to choose increased by 1 per click, and score displayed changed based on multiple selected. Once at 9 doesn’t increase any further | Multiple to choose increased by 1 per click, and score displayed changed based on multiple selected. Once at 9 doesn’t increase any further | Pass  Test Video – 0:33 |
| 3 | Multiple select change | Button decrease clicked and doesn’t decrease past 2 | left button clicked 10 times | Multiple to choose to decrease by 1 per click, and score displayed changed based on multiple selected. Once at 2 doesn’t decrease any further | Multiple to choose increased by 1 per click, and score displayed changed based on multiple selected. Once at 2 doesn’t decrease any further | Pass  Test Video – 0:33 |
| 4 | Button click | Button click to start game with selected multiple | Start game with multiple 2 selected | Game starts with multiple 2 selected, displaying score and level with 3 lives | Game starts with multiple 2 selected, displaying score and level with 3 lives | Pass  Test Video – 0::44 |
| 5 | Log creation | Log multiples have a possible solution to each layer | Game start with multiple 2 selected | Possible solution on each layer | Possible solution on each layer  Layer1 –10, 10  Layer2 – 10  Layer3 –14, 16  Layer4 – 14  Layer5 – 2 | Pass  Test Video –  0:44 |
| 6 | Frog movement | Frog moves on to river without moving onto log | Frog moves to free river selection | Frog reset, lives minus one | Frog reset, lives minus one | Pass  Test Video –  0:50 |
| 7 | Frog movement | Frog moves onto log with incorrect multiple | Selected multiple = 2, move onto log with odd numbered multiple | Frog reset, lives minus one | Frog reset, lives minus one | Pass  Test Video – 0: 57 |
| 8 | Frog movement | Frog moves onto log with correct multiple | Selected multiple = 2, Frog moves onto log with multiple of 2 | Score increase by 5, frog moves with log and isn’t reset | Score increase by 5, frog moves with log and isn’t reset | Pass  Test Video –  1:01 |
| 9 | Frog movement | Frog moves back onto log already jumped on | Selected multiple = 2, jump on correct multiple then jump off and back on again | Frog moves with log, score not increased | Frog moves with log, score not increased | Pass  Test Video –  1:11 |
| 10 | Frog movement | Frog moves to other side of the river | Frog passed the 5 layers, multiple selected = 2 | Frog reset, Log multiples changed to a different possible combination, score increased on multiple 2 by 10 | Frog reset, Log multiples changed to a different possible combination, score increased on multiple 2 by 10,  (Logs possibility’s)  Layer1- 2  Layer2-12, 2  Layer3- 12  Layer4- 18  Layer5- 4 | Pass  Test Video –  1:21 |
| 11 | Frog movement | Frog moves off left or right of the screen | Player jumps on first row a waits to move right off the screen | Life lost and frog re-set | Life lost and frog re-set | Pass  Test Video –  1:56 |
| 12 | Level | Score achieved to change level | Score achieved >= 150 | Level changed to level 2, Log speed increased by 1 | Level changed to level 2, Log speed increased by 1 | Pass  Test Video –  2:10 |
| 13 | Lives | Lives left = 0 | Frog dies 3 times | Takes to pause screen | Takes to pause screen | Pass  Test Video –  2:25 |
| 14 | On exit | On exit scores saved | Exit clicked –  Student logged in Test Test | Score currently achieved saved to backend to student who has logged in | Score currently achieved saved to backend to student who has logged in | Pass  Test Video – 2:30 |
| 15 | Scores saved | Scores saved to student with same name as an already existing student | Exit clicked –  (Student logged in)  First name -Test  Surname – Test  StudentId - 2  (Another student)  First name -Test  Surname – Test  Student Id - 1 | Data saved to student with Id - 2 | Data saved to student with Id -1 | Pass  Test Video –  2:40 |

## Demonstration – Black box testing

## Form1

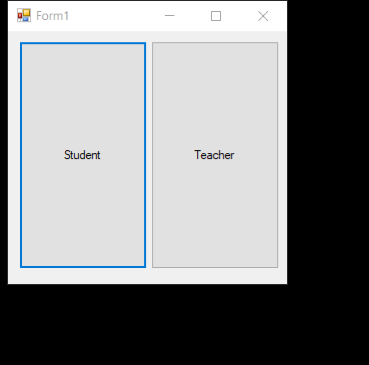
### Test 1

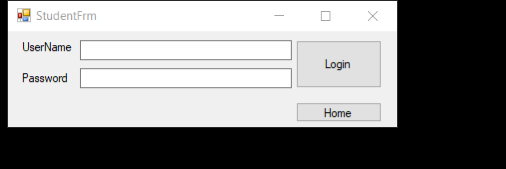
On click of Teacher button, Form 1 is closed and LoginFrm is displayed



## Test 2

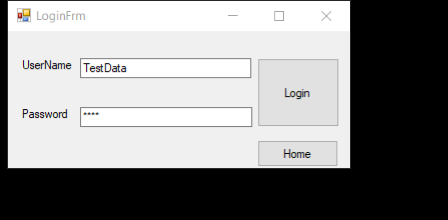
On click of student button Form 1 is closed and StudentFrm is opened.



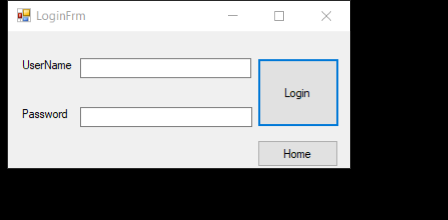
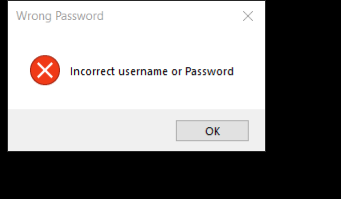


## Login Form

### Test 1



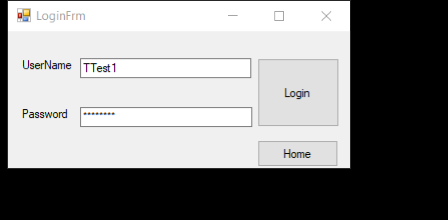
Incorrect username input

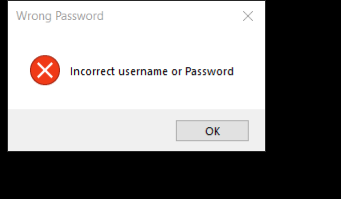


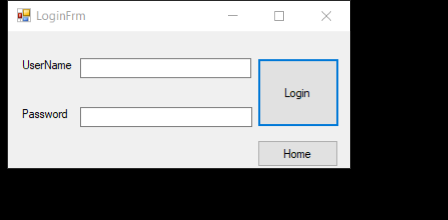
Message box displayed

Username and password inputs cleared, allows for another attempt

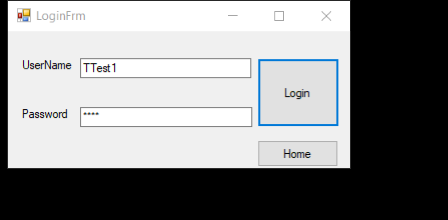
### Test 2

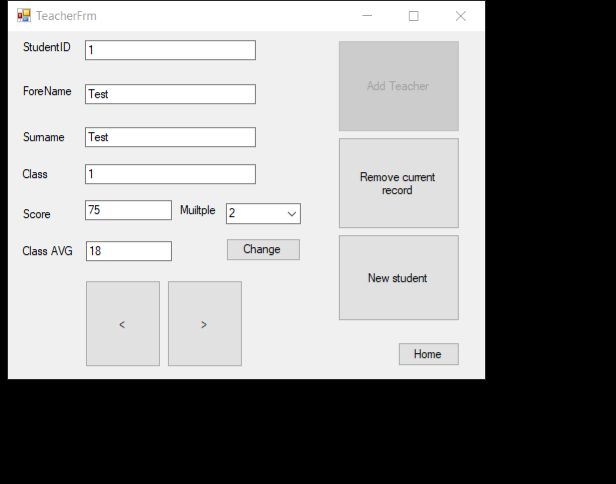
Incorrect password (TestData)

Message box shown on attempted login

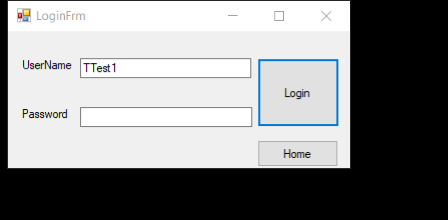
Clears input boxes to allow for another attempt

### Test 4

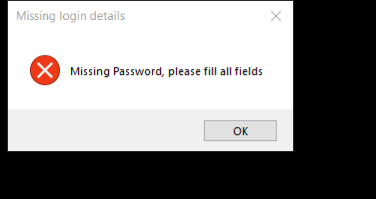
Correct username and password input

Login form closed and Teacher form opened, logged into teacher Test Test, who’s class in number 1

### Test 5

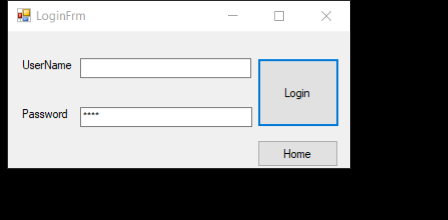


Input username missing password

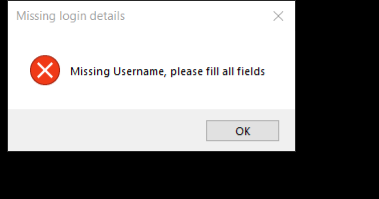


Message box shown prompting user to fill all fields

### Test 6

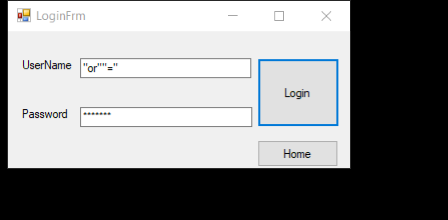


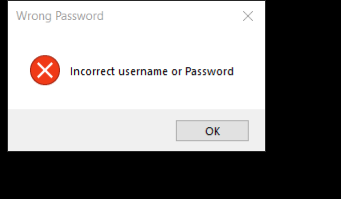
Password input, missing username

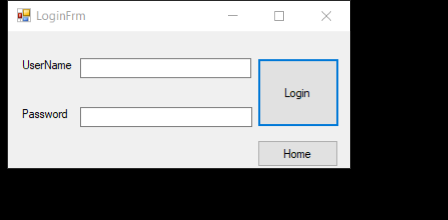


Message box displayed prompting user to fill all fields

### Test 7

Sql injection statement

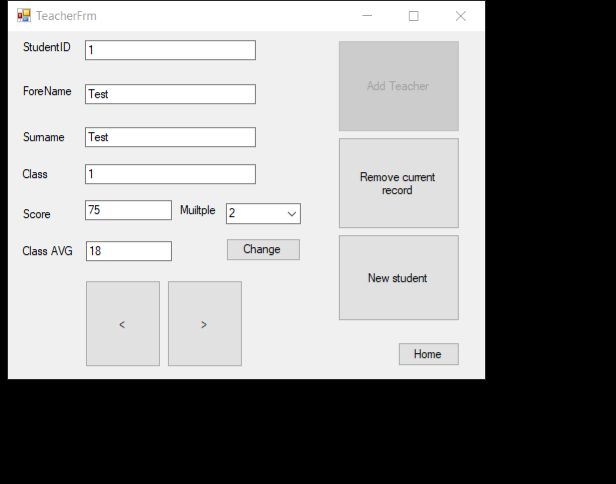
message box displayed



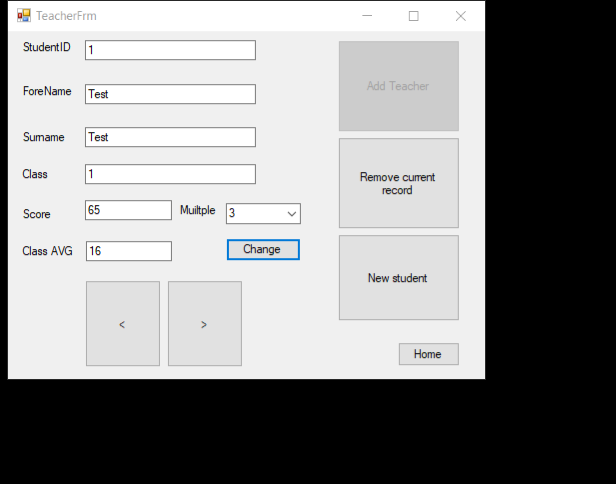
prevention from login

## Teacher form

### Test 1

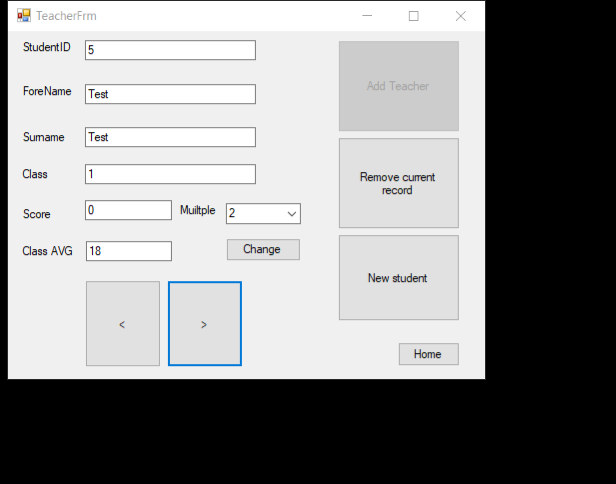


Multiple displayed changed



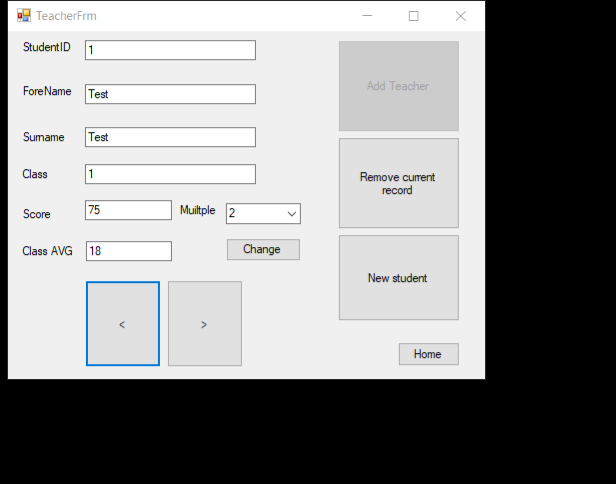
Score and average score for multiple 3 displayed

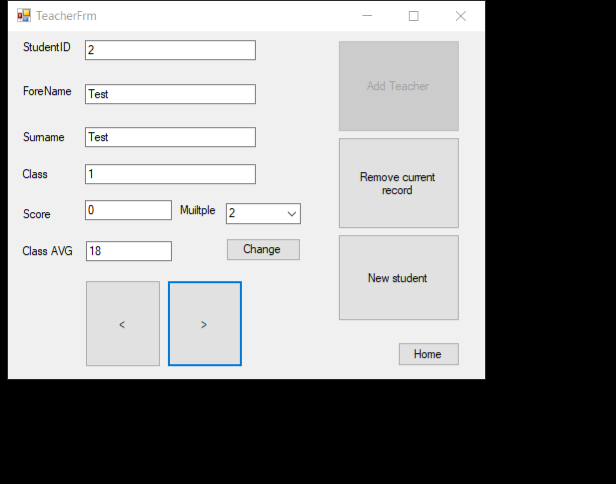
### TeacherFrmTeacherFrmTeacherFrmTest 2



Currently 4 students in class 1, Teacher – Test Test, TeacherId – 1, class -1 is logged in only students in class 1 have been displayed.

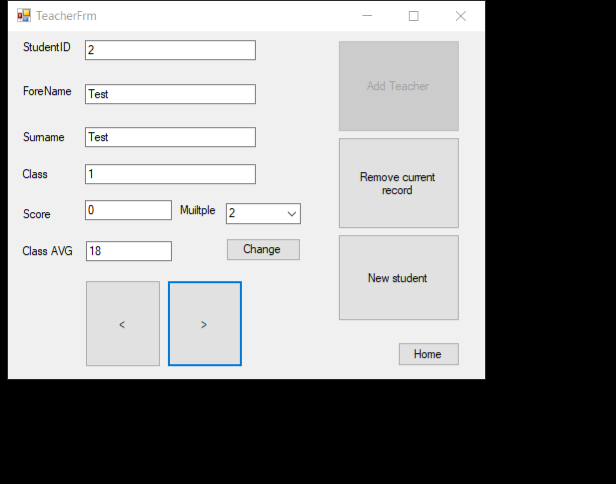
### Test 3

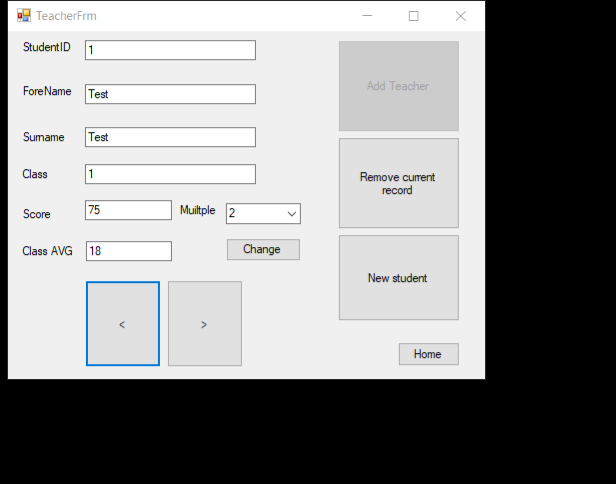


Current student displayed StudentID – 1, Right button clicked student displayed, StudentID - 2.

### Test 4

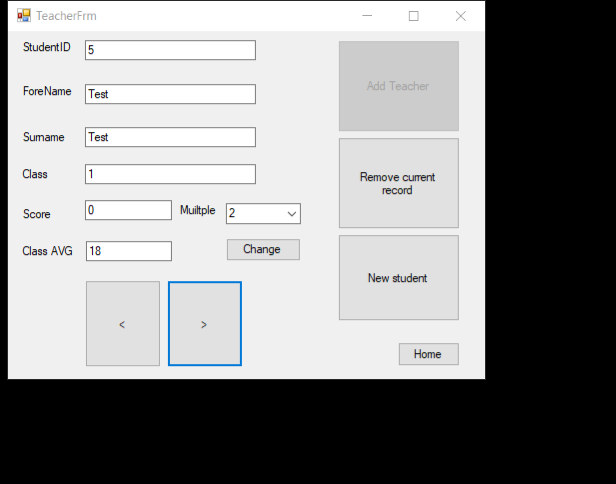
Like test 3 just the inverse



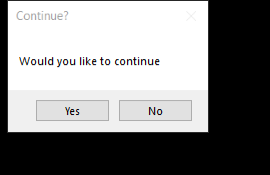


Current student displayed StudentID – 2, left button clicked student displayed StudentID -1

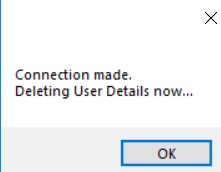
### Test 5



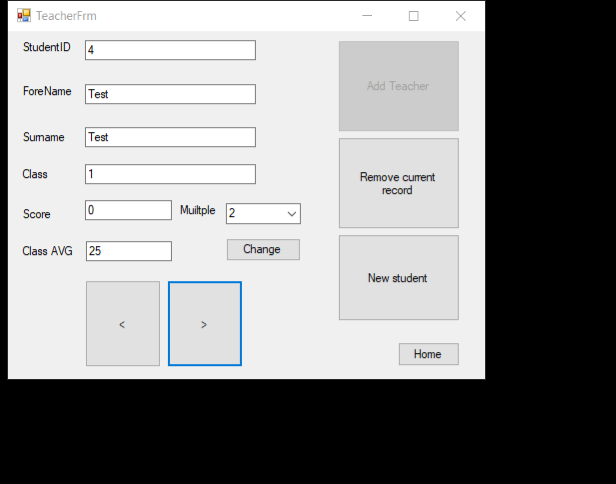
Currently selected record StudentID – 5, remove record button clicked



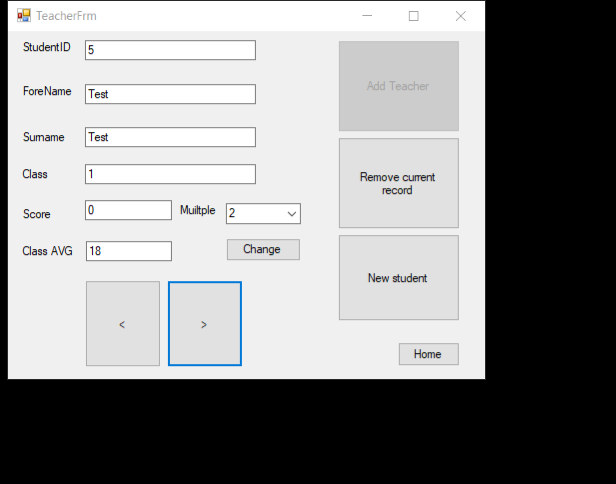
Continue message box displayed. Yes clicked.



Second message box displayed confirming user has been deleted.

Form reloaded with student StudentID - 5

### Test 6



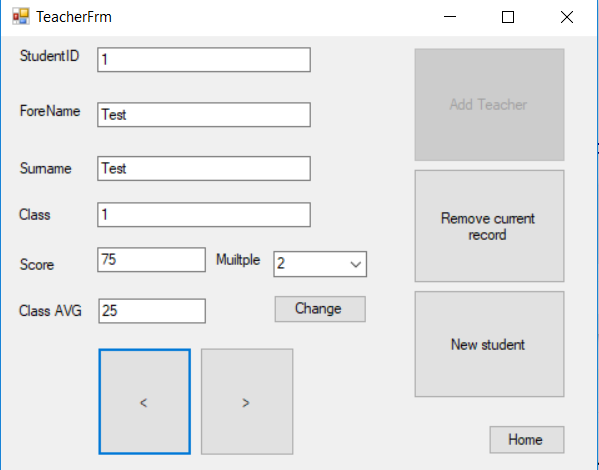
before student StudentID - 5 was removed

Multiple selected – 2

Number of students – 4

Total score – 75

Class average – 75/4 = 18.75 - truncates



After student StudentId -5

Multiple selected – 2

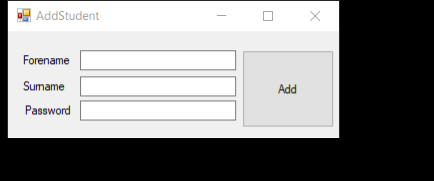
Number of students – 3

Total score – 75

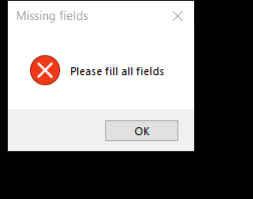
Class average – 75/3 = 25

## Add Student

### Test 1

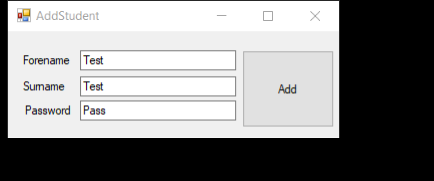


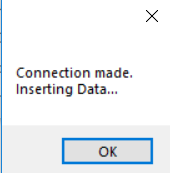
Fields missing

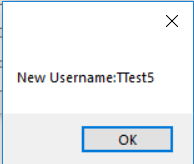


Message box prompting to fill all fields

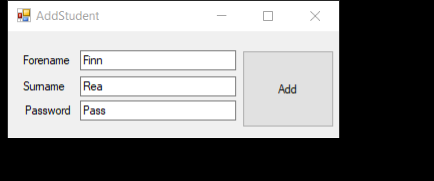
### Test 2

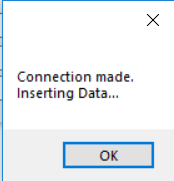
Currently there have been 4 other students with First and surnames, Test Test, this one should create a username TTest5

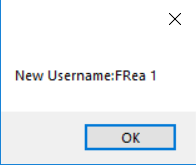


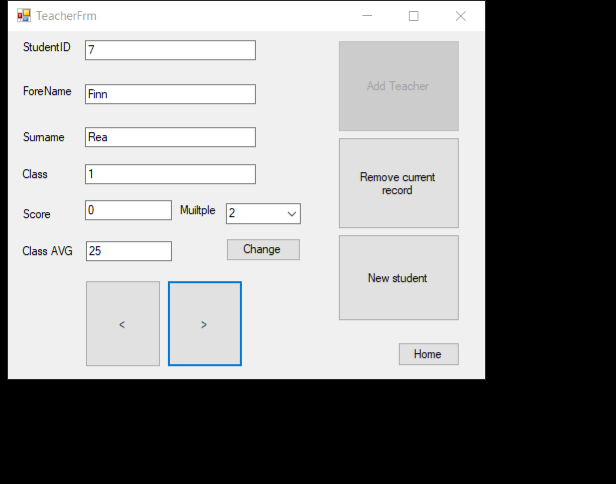
 displays new username

### Test 3

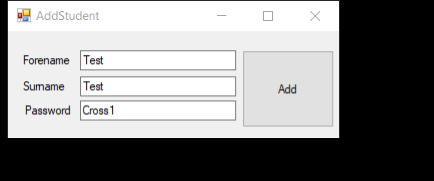
 All relevant information for new student



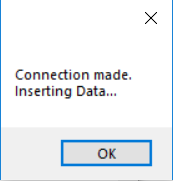
 New student create demonstrated, with Username – Frea1, Password – Pass

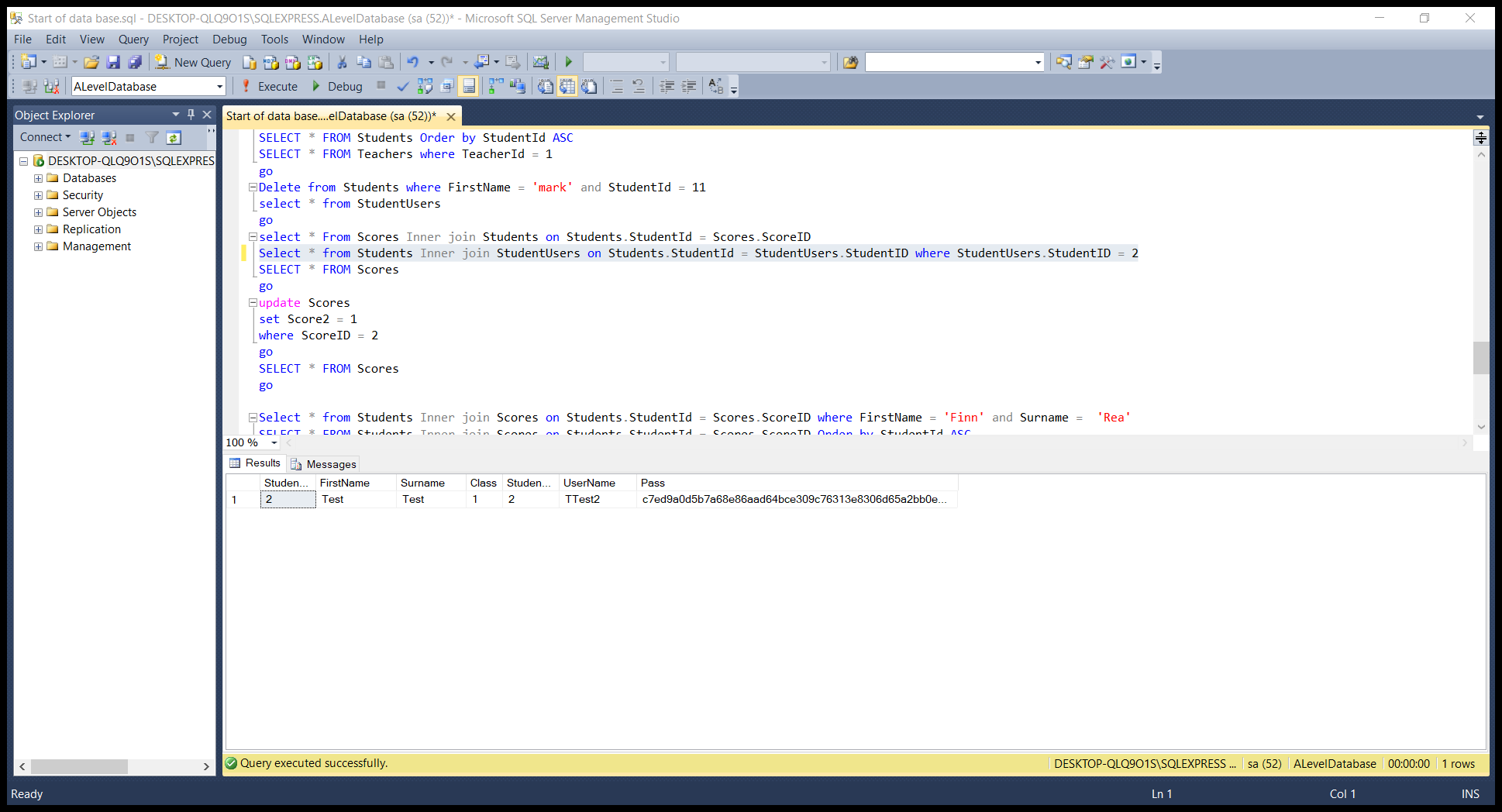
 New record

### Test 4



New student created Username – TTest2, Password – Cross1

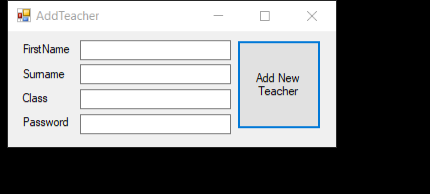


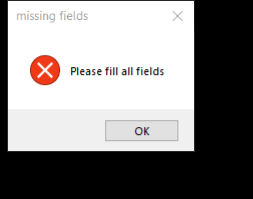


Sql statement and demonstration of password being hashed

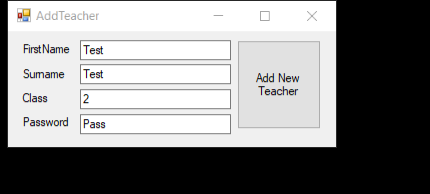
## Add Teacher

### Test 1

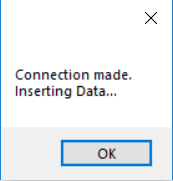
All fields missing

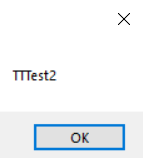
Message box displayed prompting user to add missing fields

### Test 2



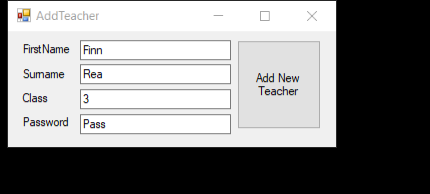
Already exists a username from first name – Test, surname – Test, Username – TTTest1. New username with these fields should be - TTTest2, Added T on the front in case a student would have the same username.



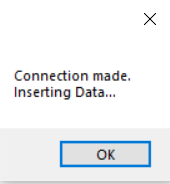


New Teacher created with Username – TTTest2, Password – Pass

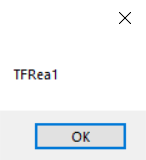
### Test 3

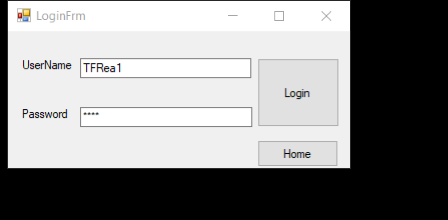


New Teacher created, First name – Finn, Surname -Rea, Class – 3, Password – Pass. Username created should be – TFRea1, Password - Pass

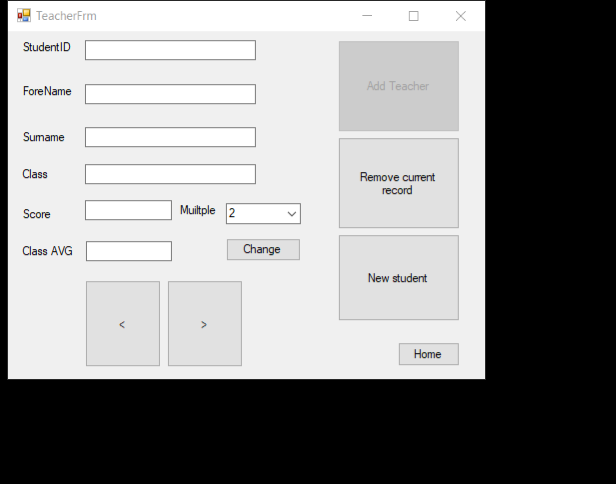


Demonstration created

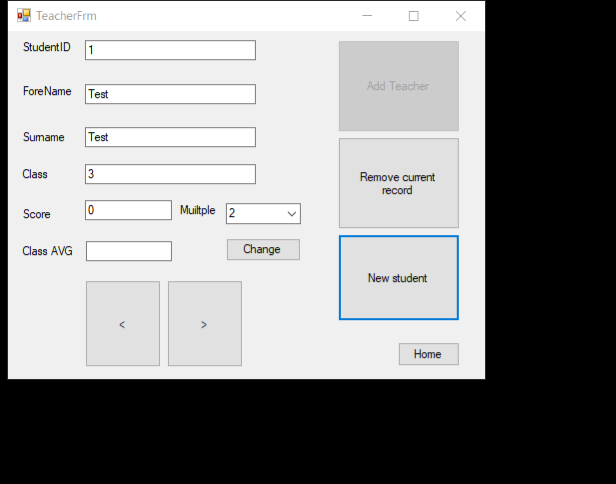


 Displayes username created

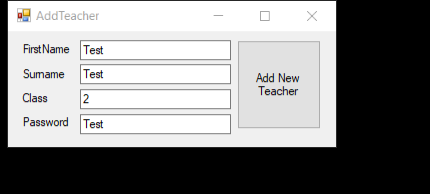
Attempt of Login with Username and Password created.

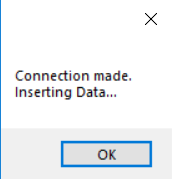


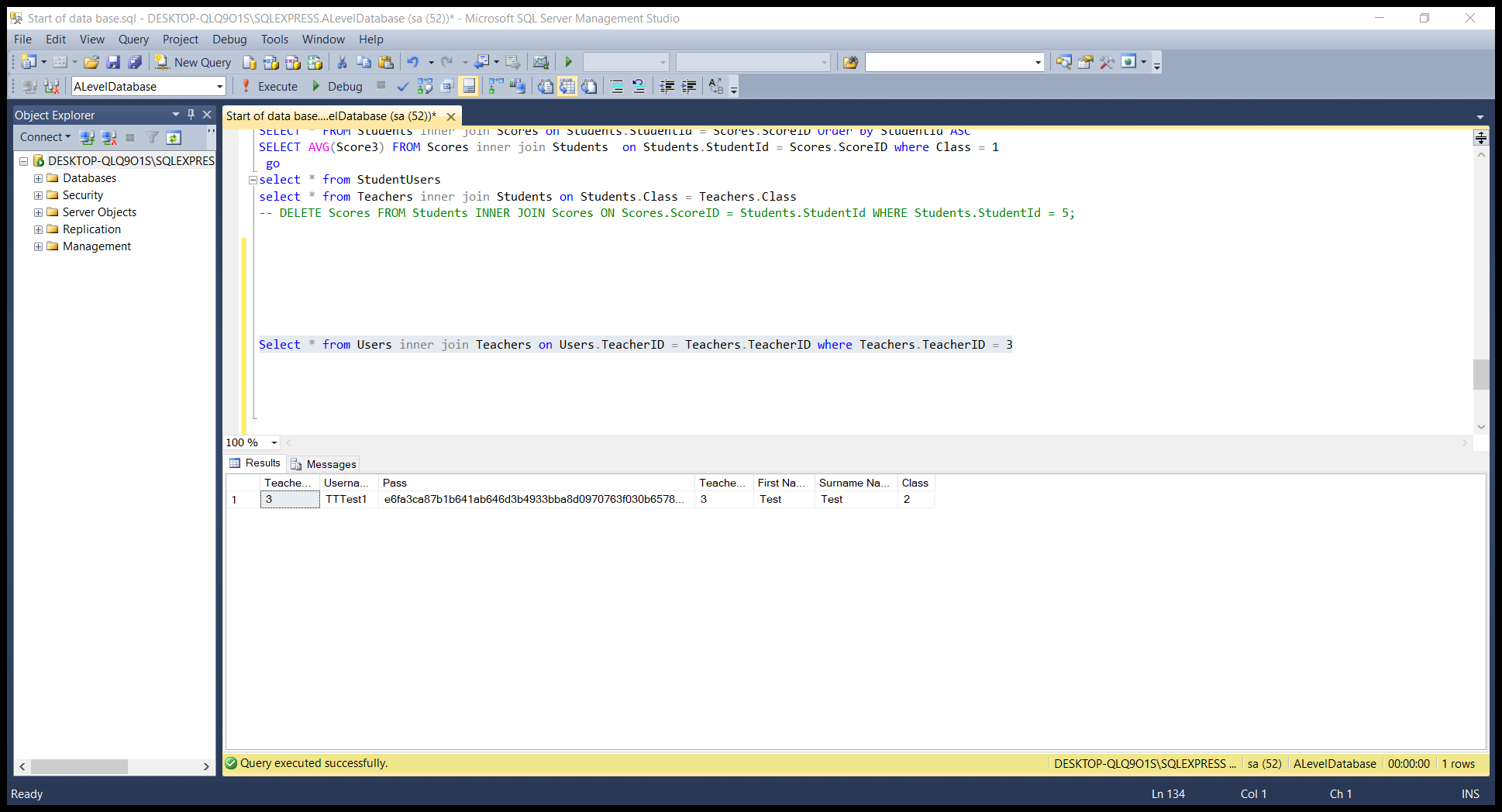
Shows Teachers class, of whose logged in

New student added into class 3.

### Test 4







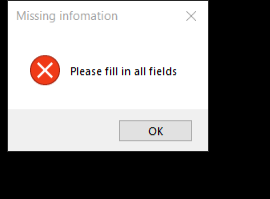
Stored password is hashed

## StudentFrm

### Test 1



Missing Username



Requests user to input information into all fields

### StudentFrmTest 2

Incorrect Password

Message box displayed showing incorrect login

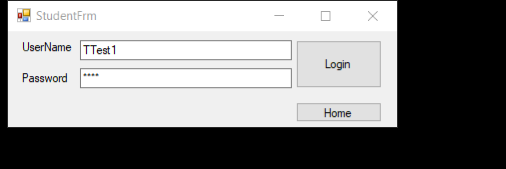
### StudentFrmTest 3

Incorrect username

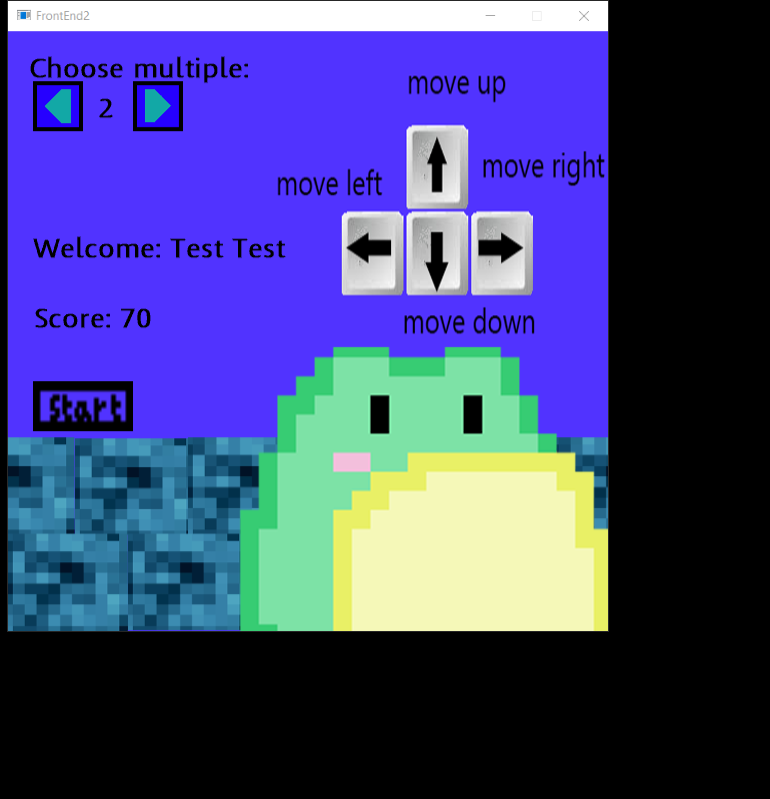


Login prevented message box displayed showing incorrect login

### Test 4



Correct Login input



Game is opened welcomes the student Test Test (which has just logged in).

### Test 5

Current Scores for Test test, StudentID - 1

Score 2 – 70

Score 3 - 0

Score 4 - 0

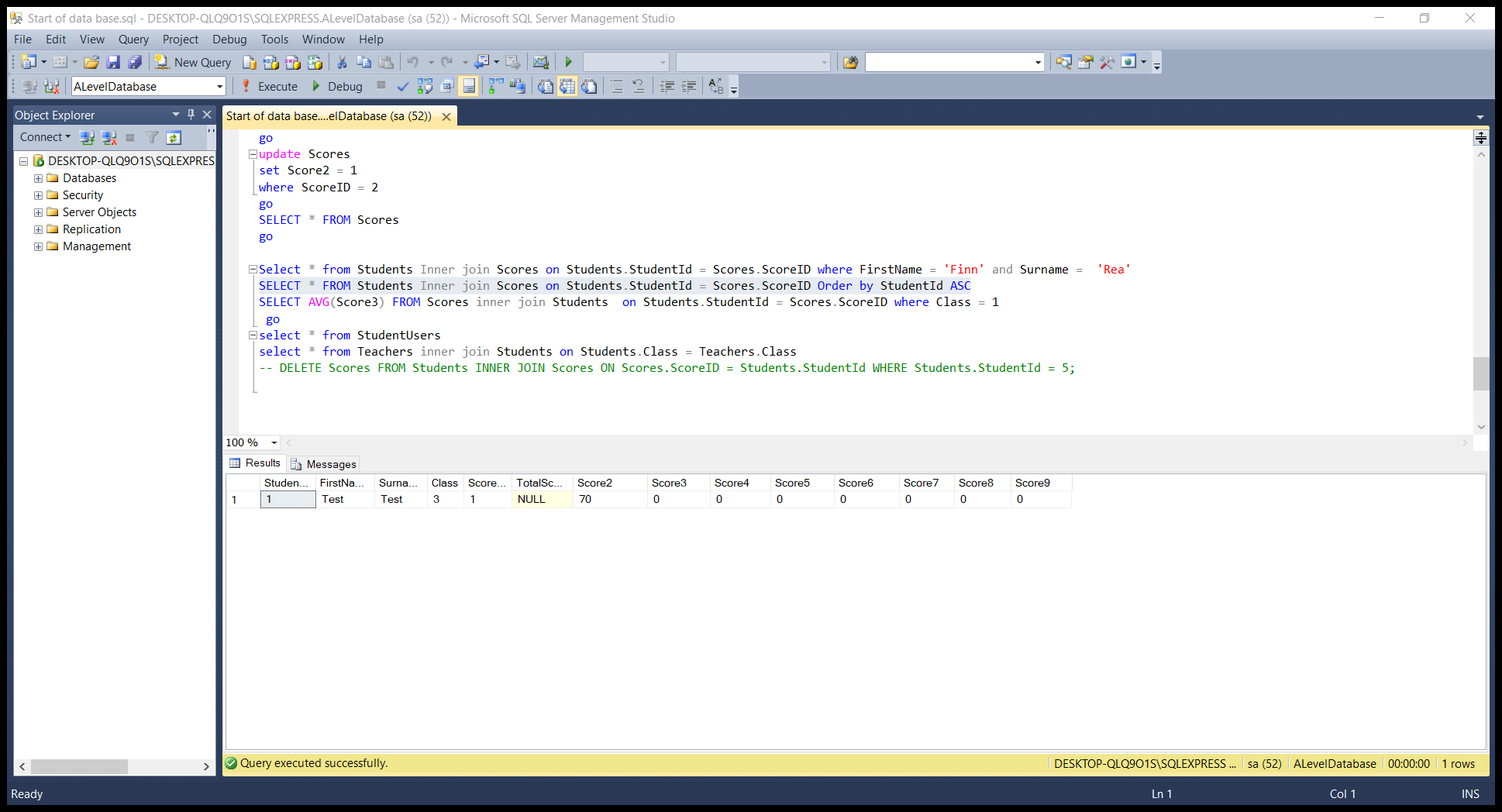
Score 5 - 0

Score 6 - 0

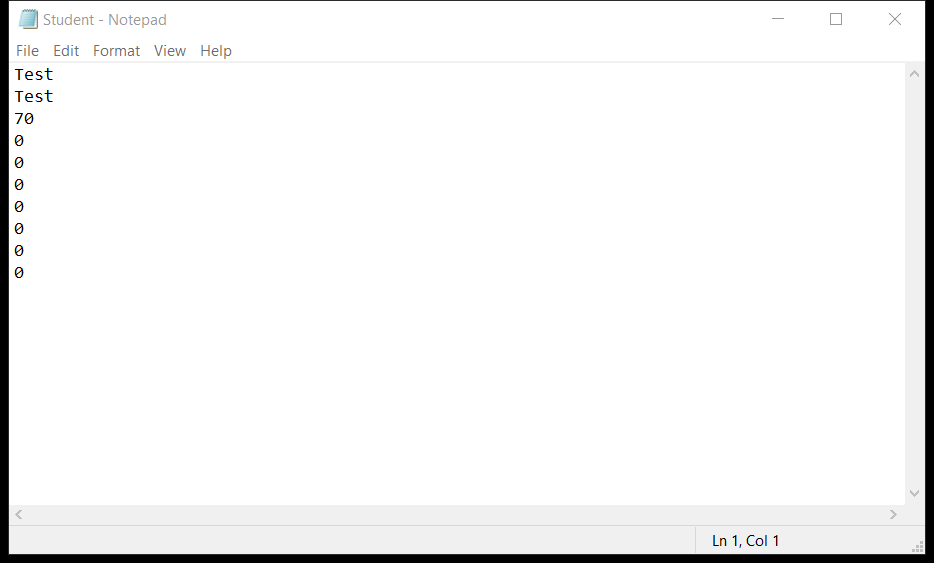
Score 7 - 0

Score 8 - 0

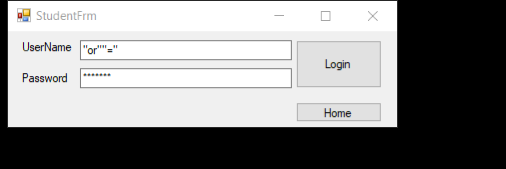
Score 9 - 0



Text file –



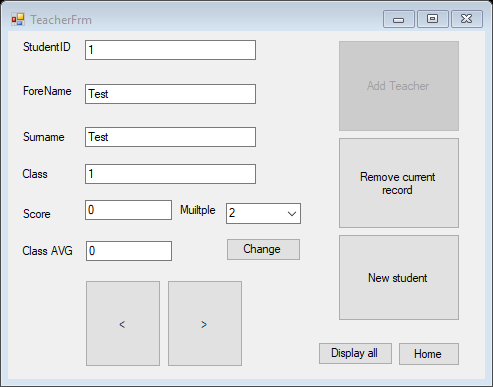
### Test 6



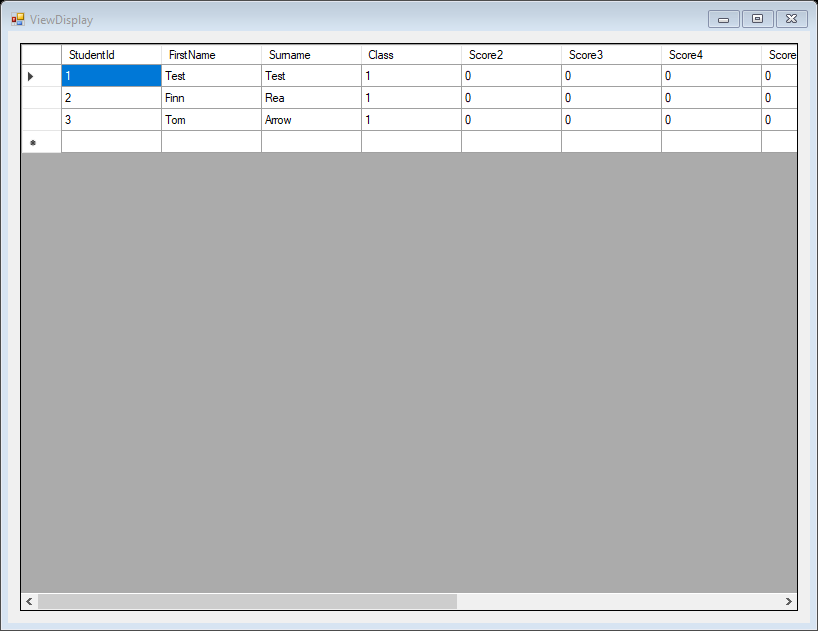
Attempted SQL injection

 Incorrect Login message box shown

### VeiwdisplayForm



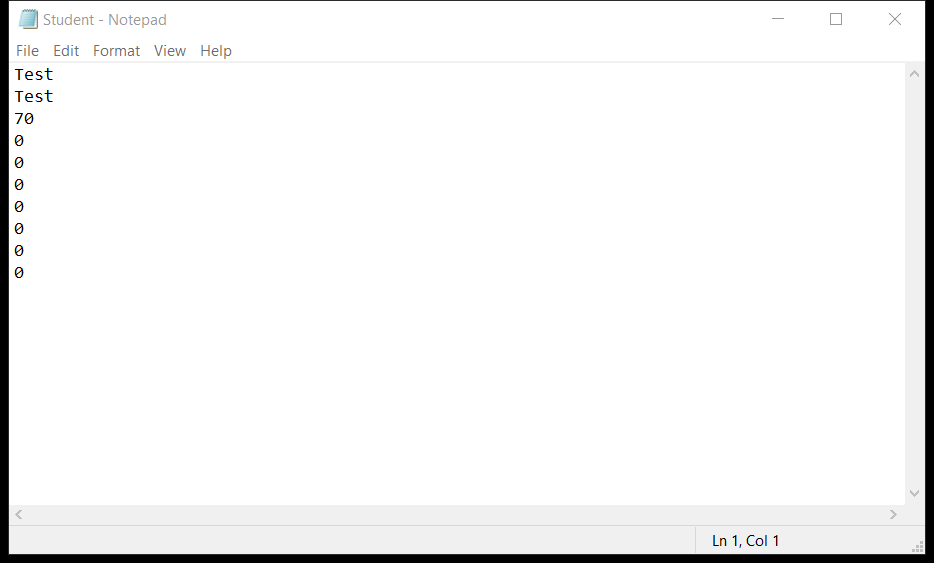
Logged in for teacher with class 1



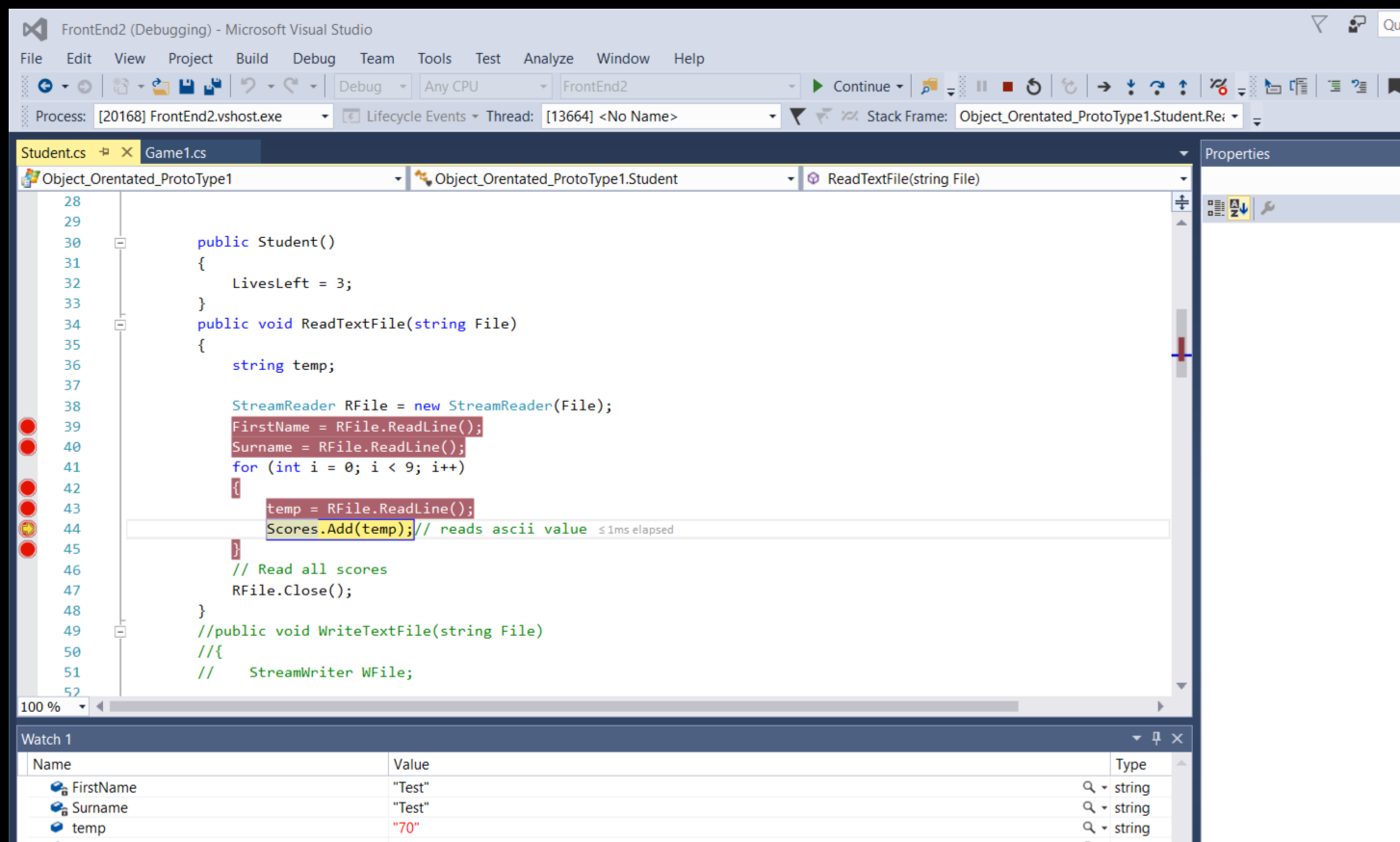
All students displayed in class 1

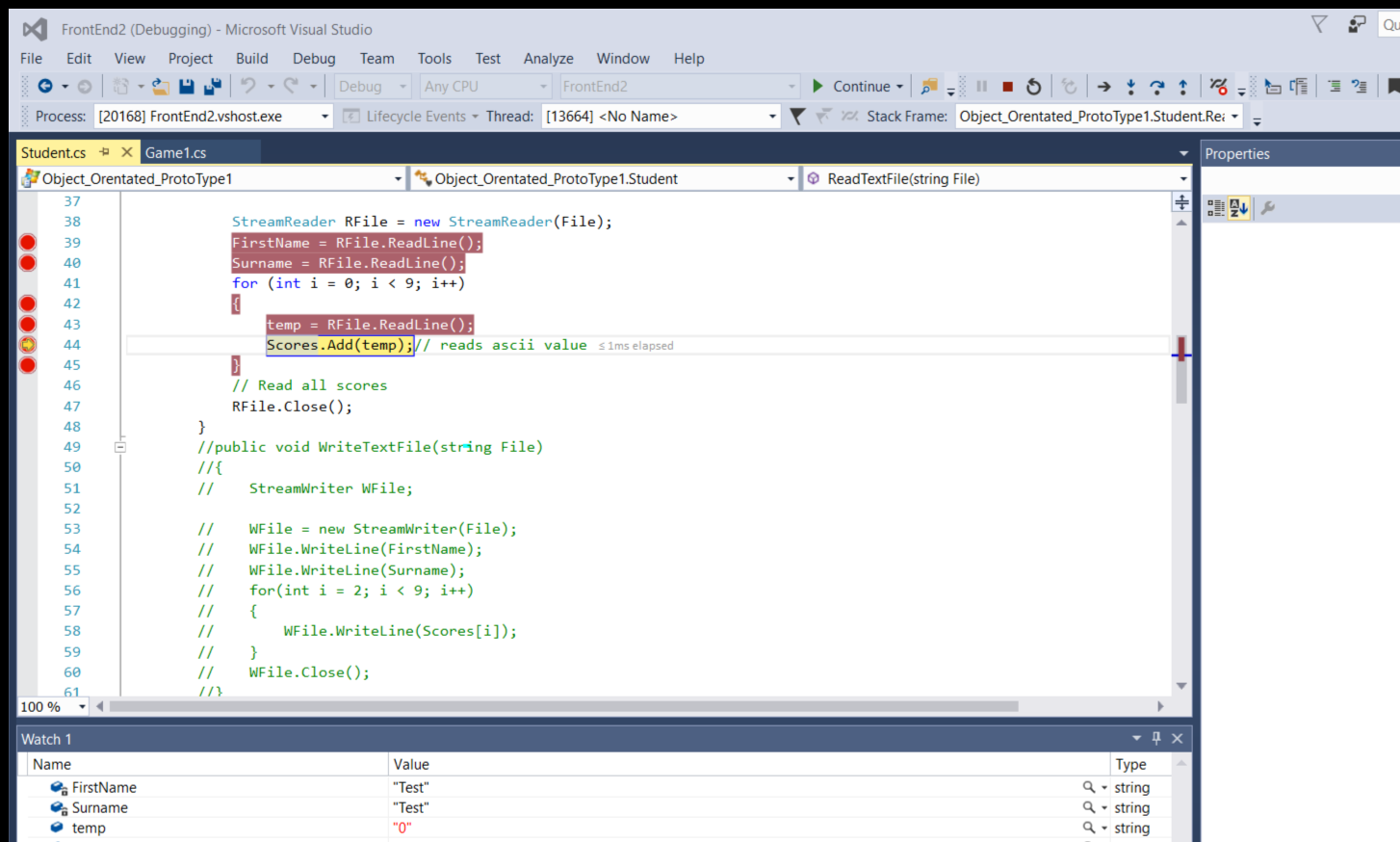
## Game – White box testing

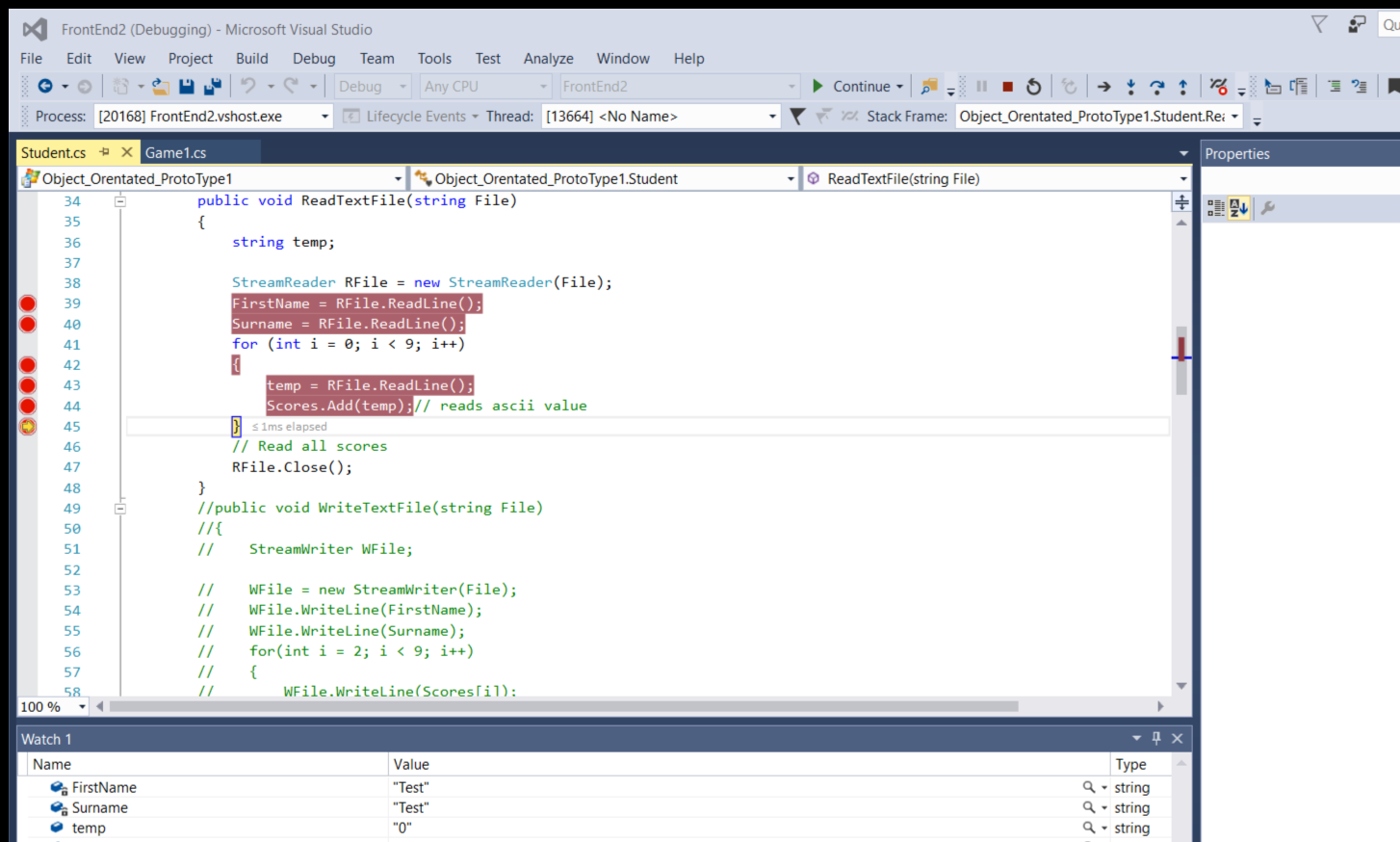
### Test 1

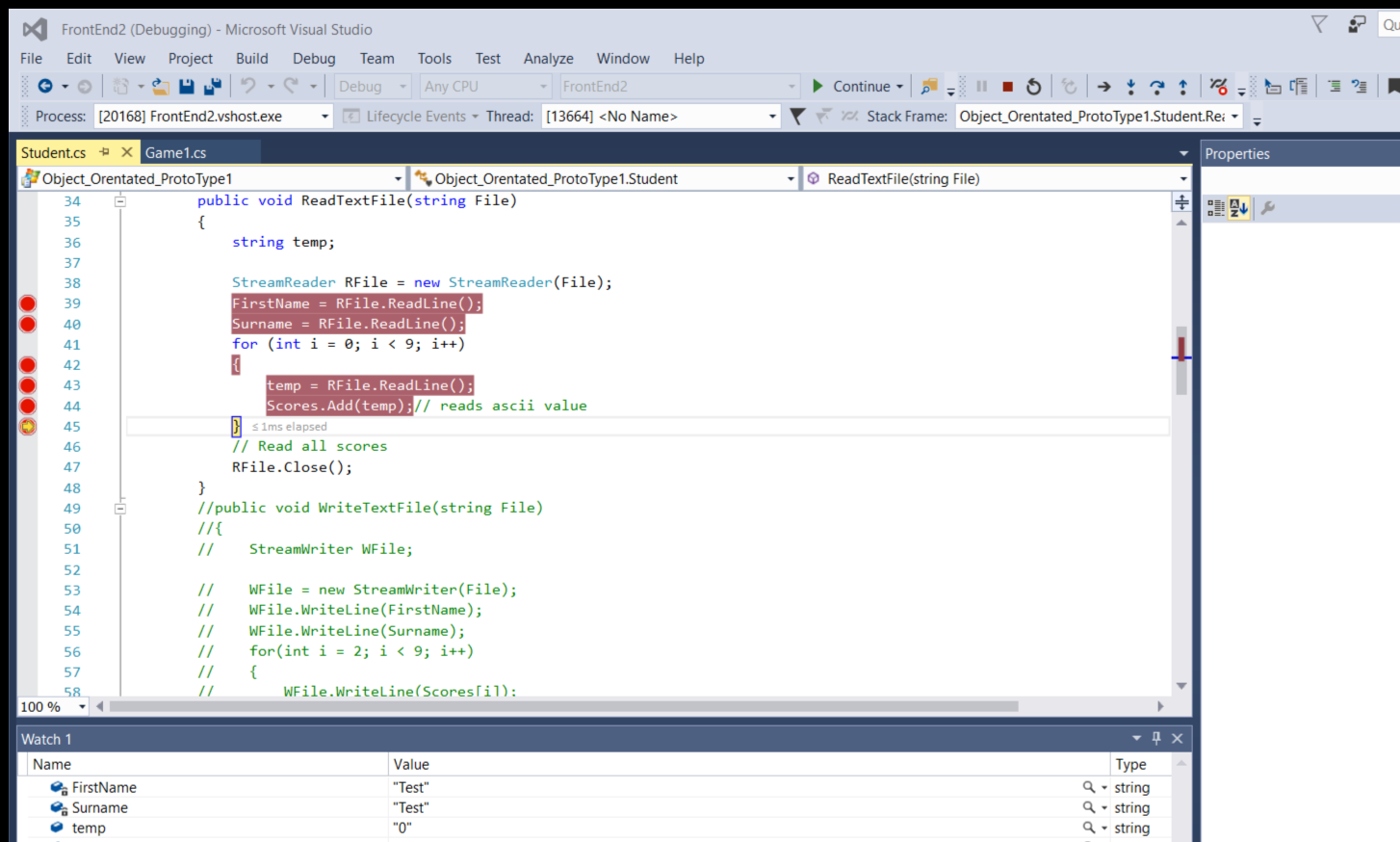


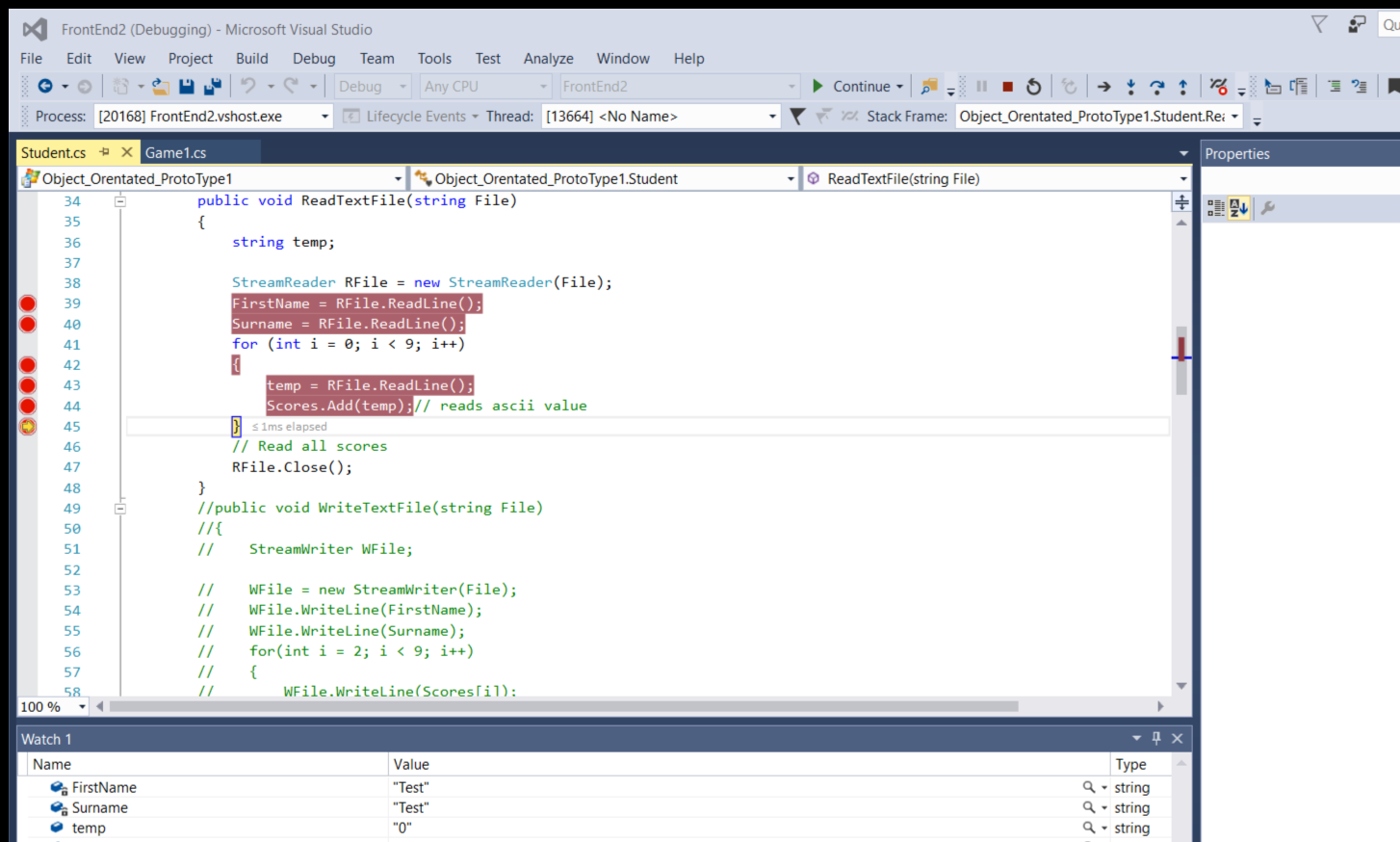
Current text file expect to read first name = “Test”, Surname = “Test”, temp = 70, 0, 0, 0, 0, 0, 0, 0, changes each iteration, and adds into score list. Score list expected to count 8, with the same numbers added.

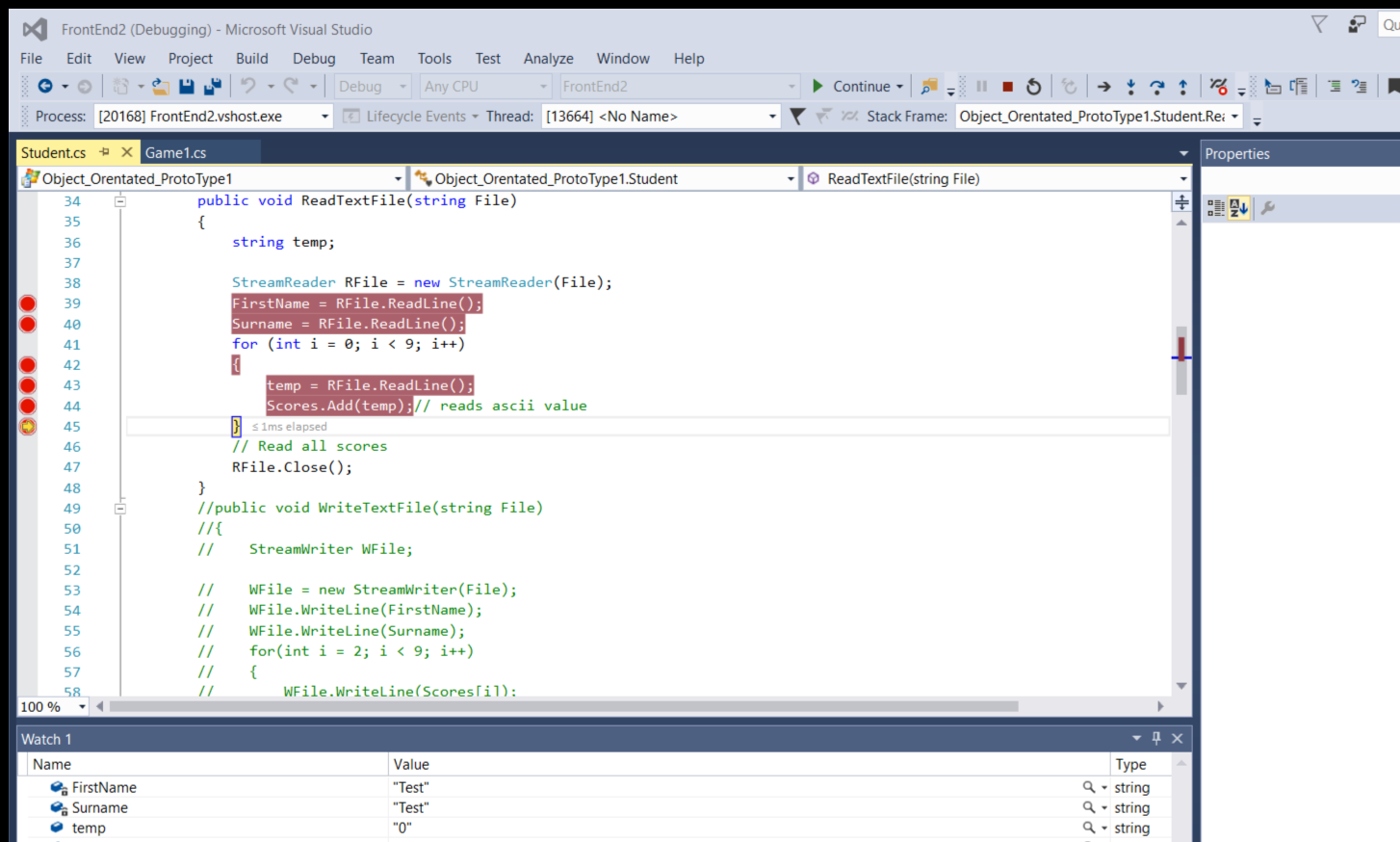


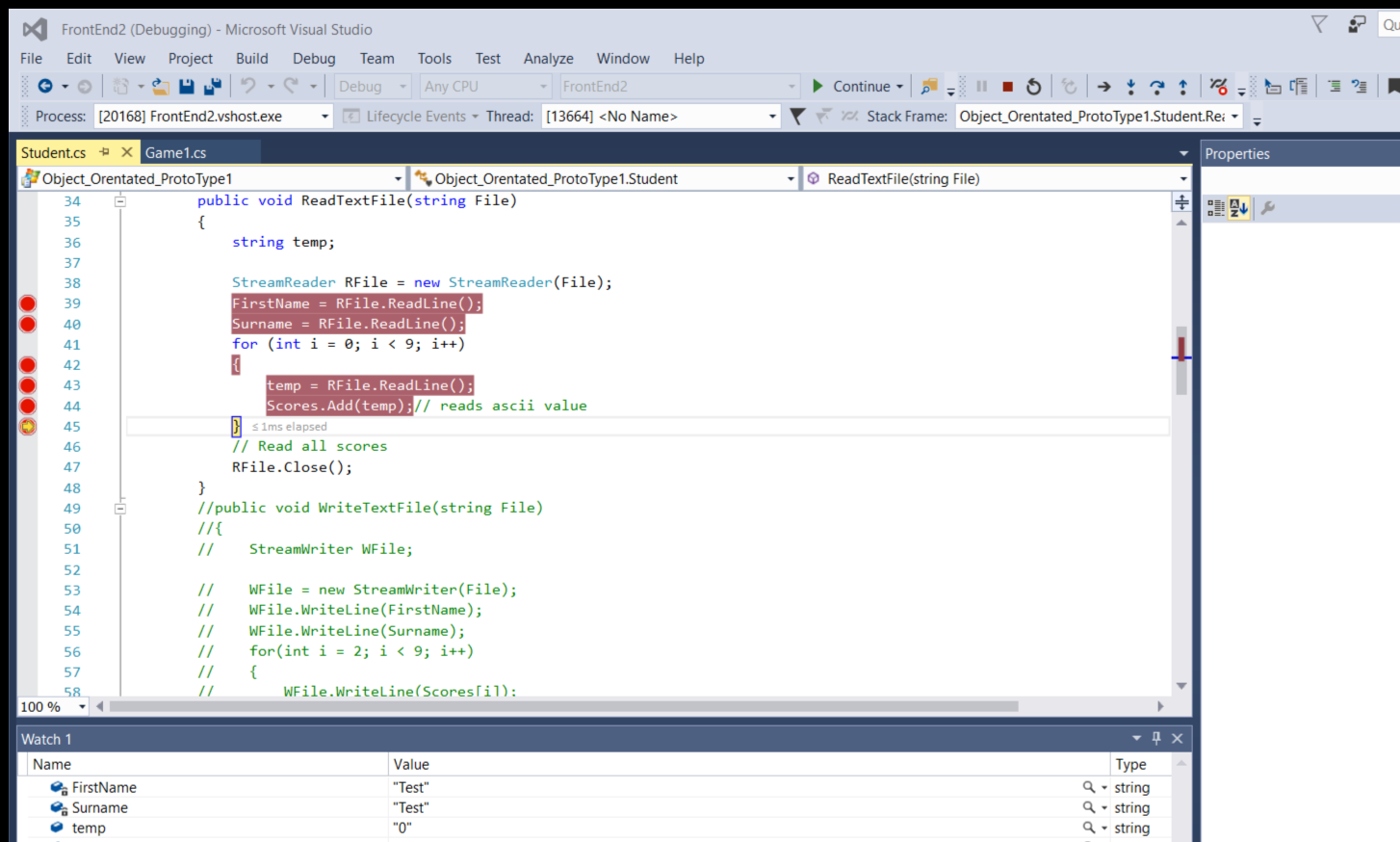


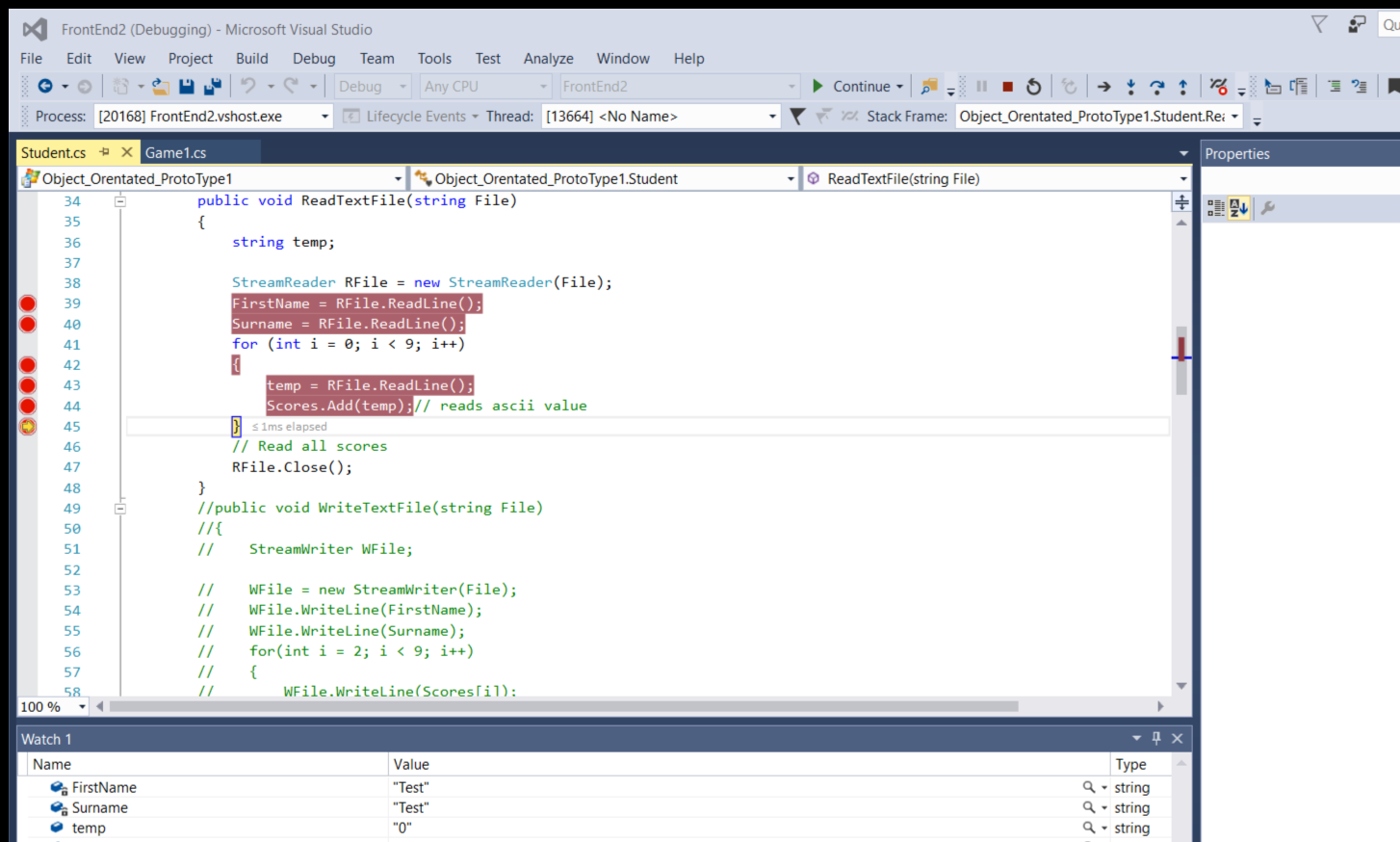


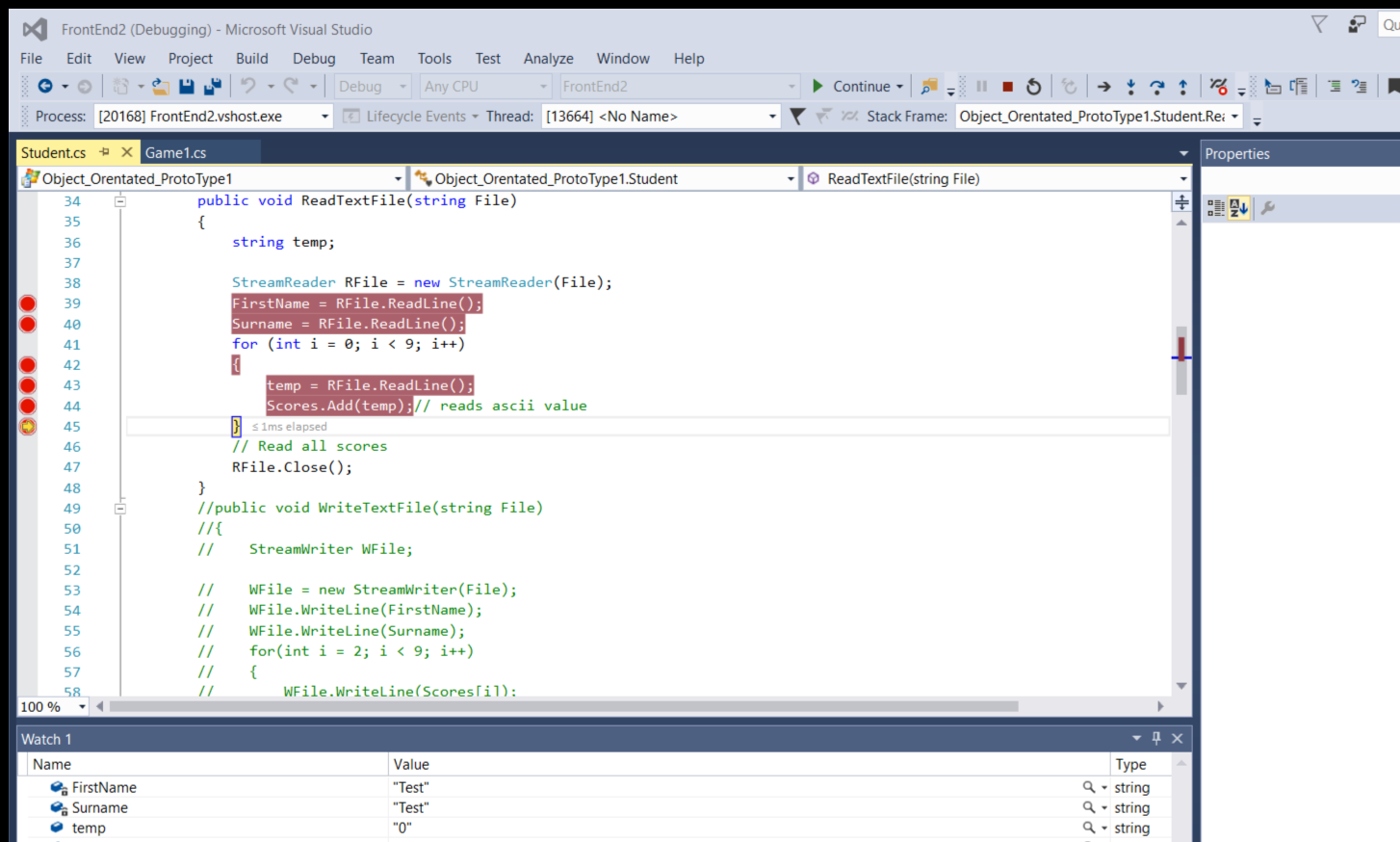


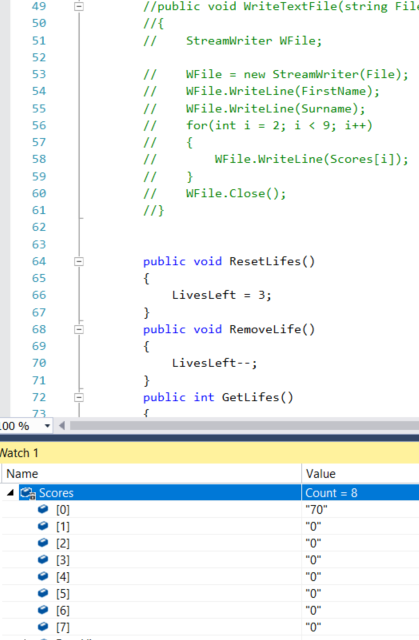






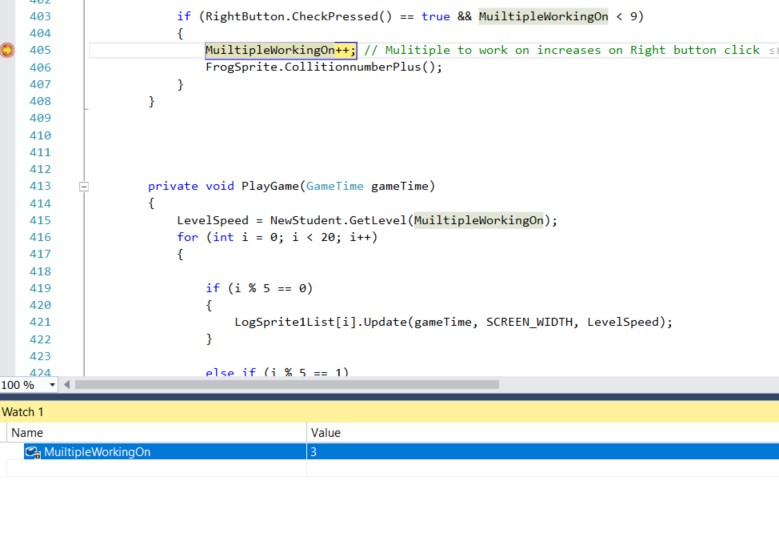






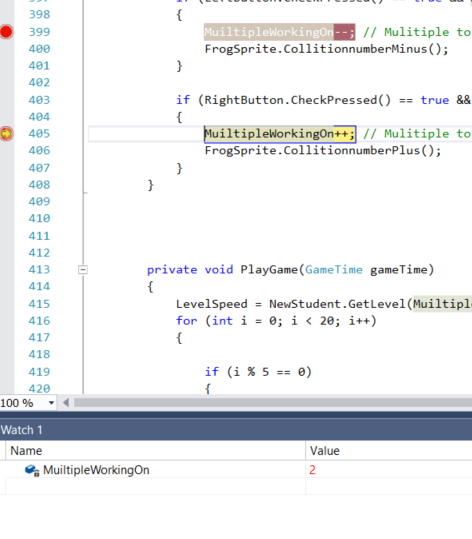
All numbers added to score correctly.

### Test 2

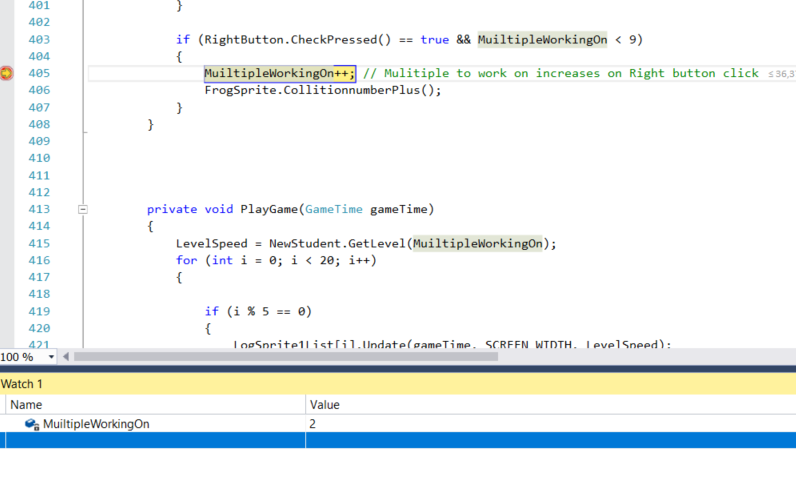


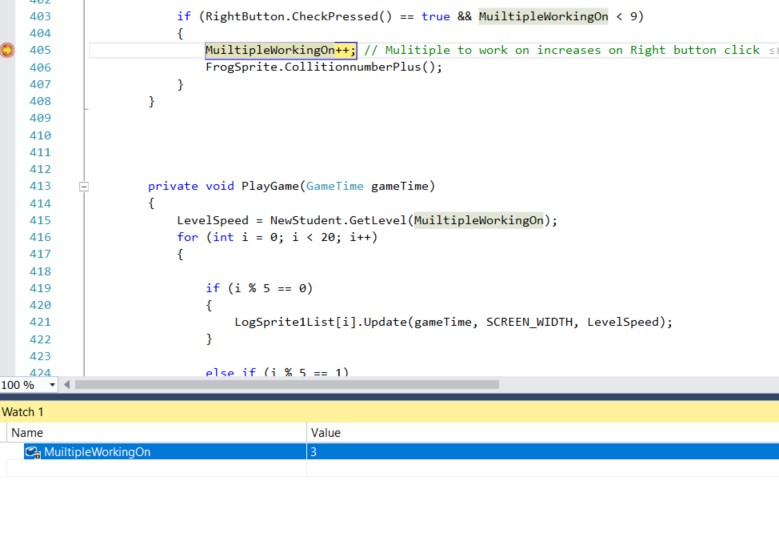
Current multiple selected is 3

Left button clicked

Current multiple selected changed from 3 => 2 (Shows increasing as the only way to show change is to re-trigger the break point)

### Test 3

Current multiple selected = 2

Right button clicked

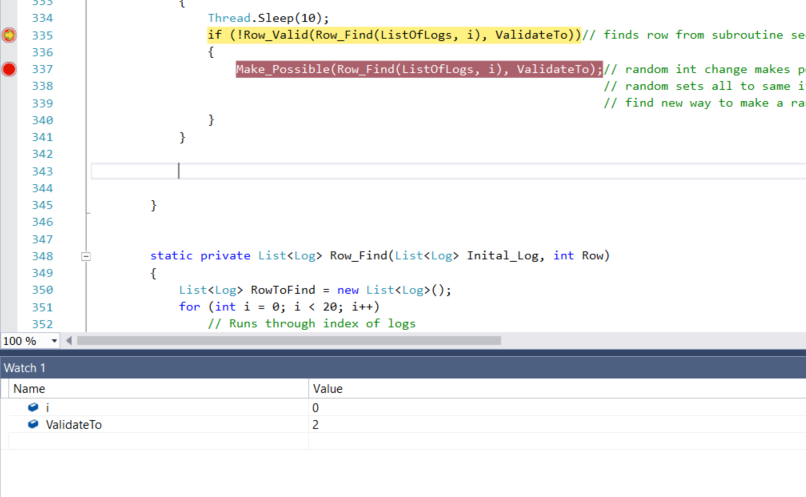
Current multiple selected changed from 2 => 3

### Test 4

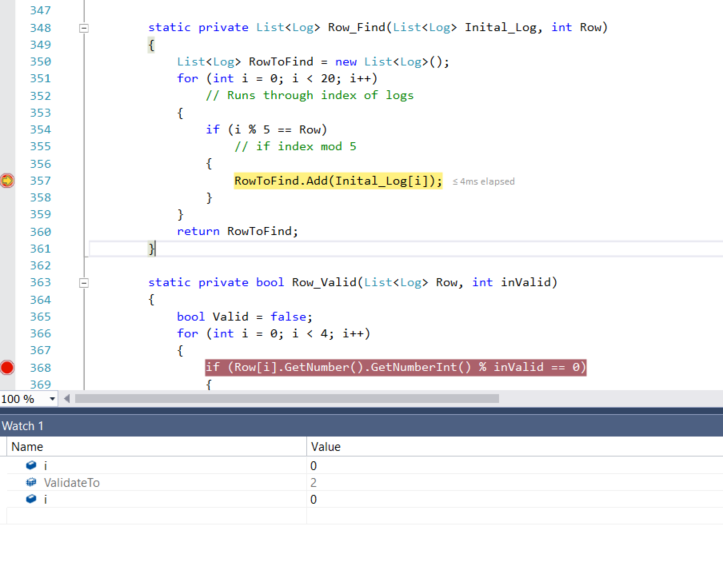
Unable to break point this as it is in update

### Test 5

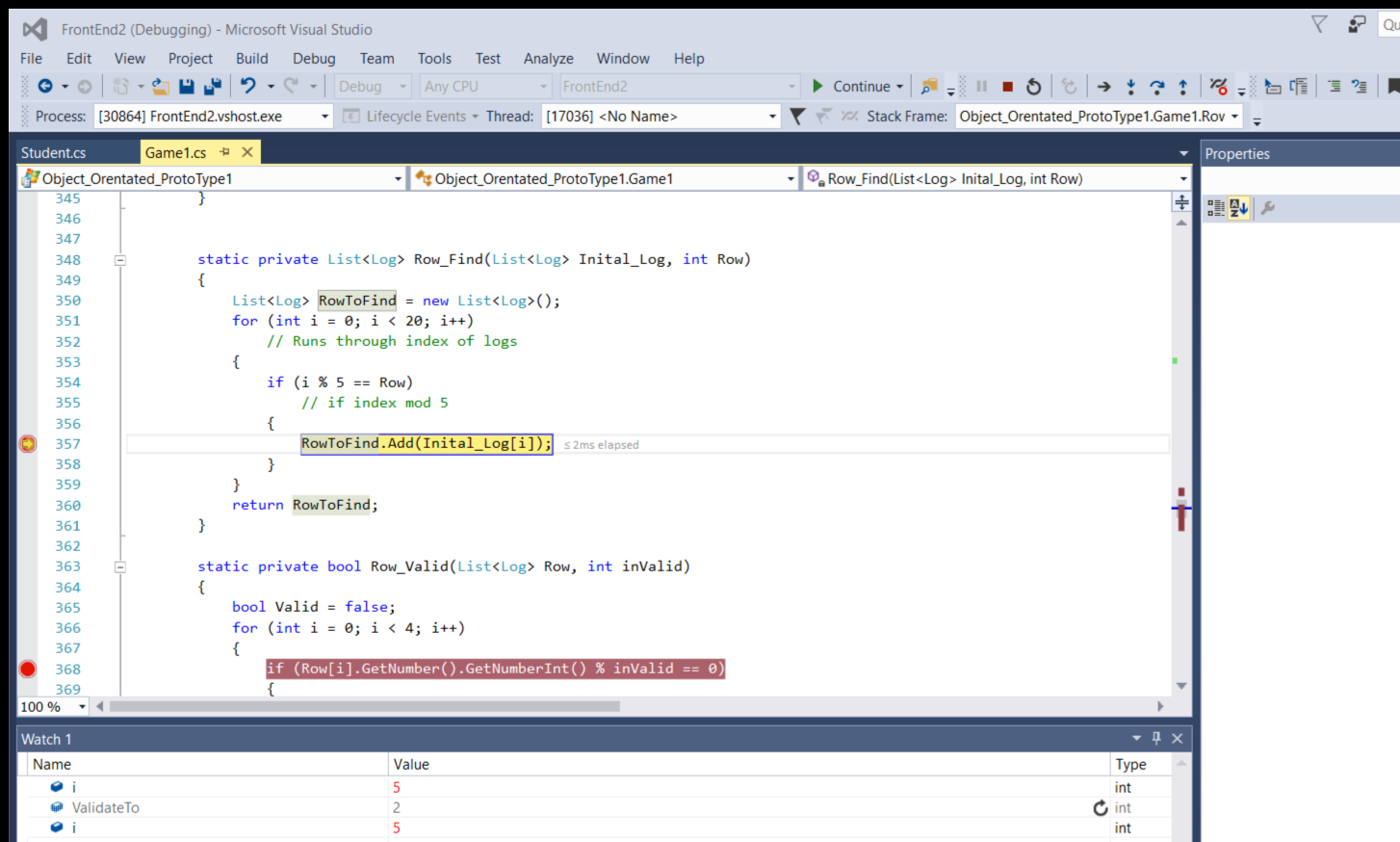
Start button clicked, multiple 2 selected

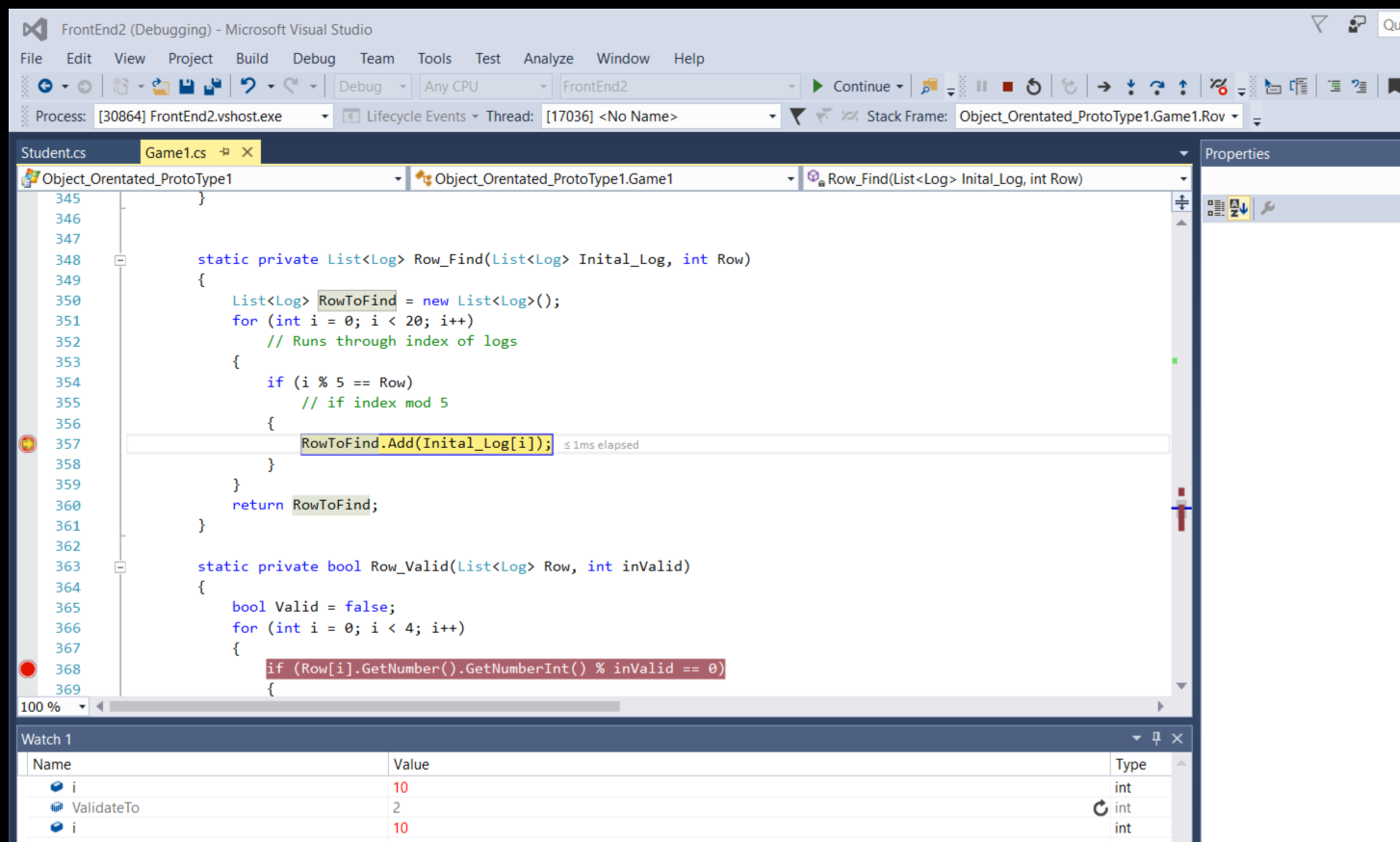


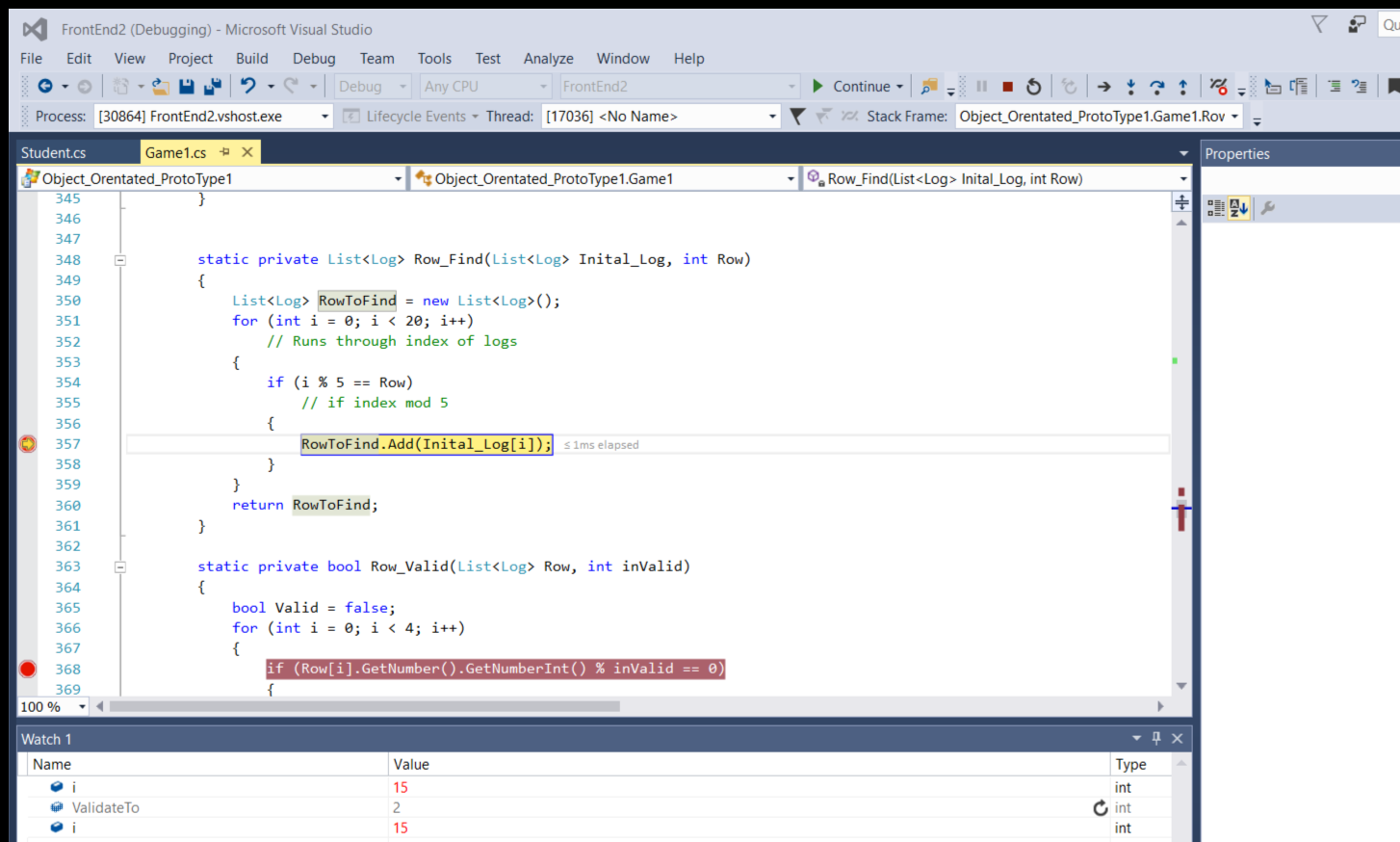
I = Row, I at 0 means check row 1, validate to = multiple selected

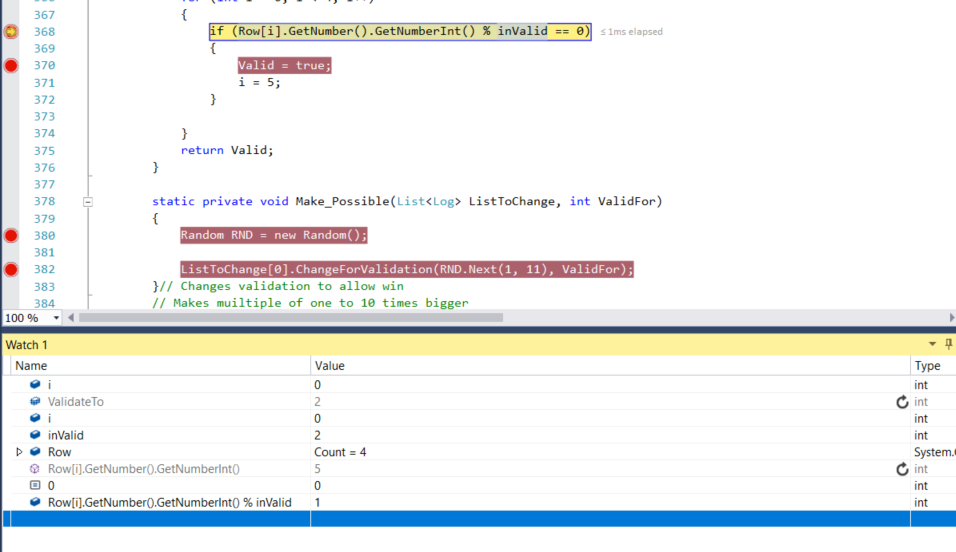


Takes to row find subroutine, adds all objects that index mod 5 = row to find which is currently 1



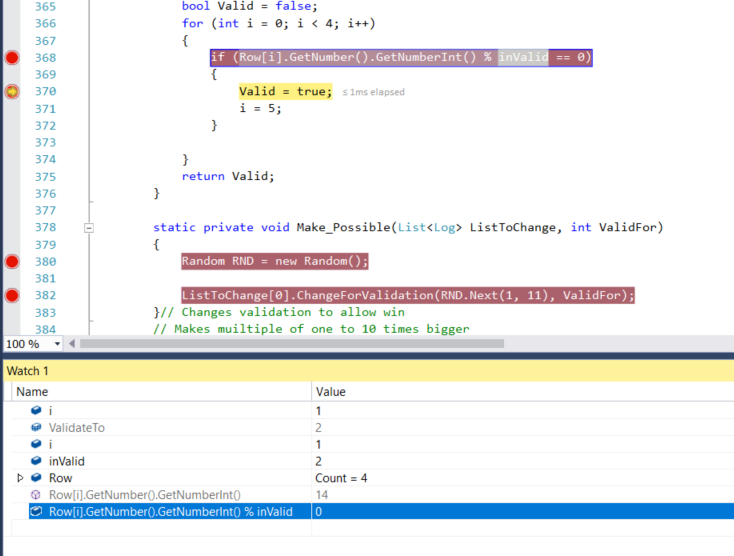






Log 1 - Not valid Log number (5) mod 2 = 1 not 0

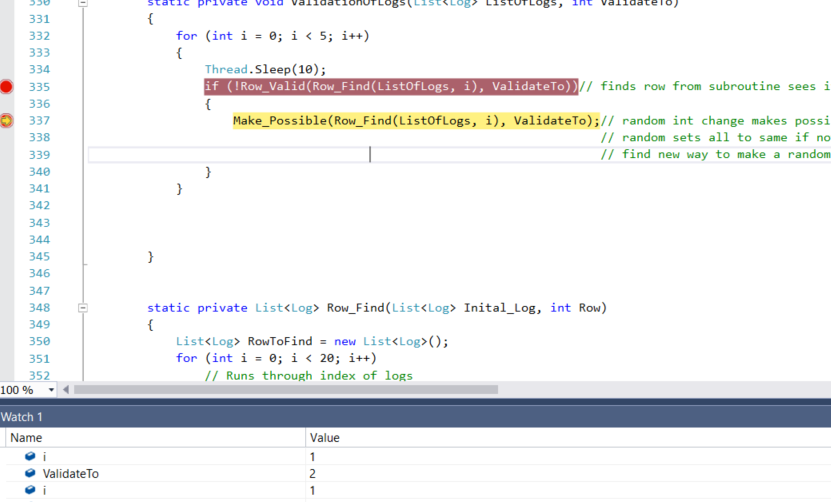
Then row that was found is taken to check valid, this iterates over each object in the list checking if its multiple is a factor of the chosen one by the student



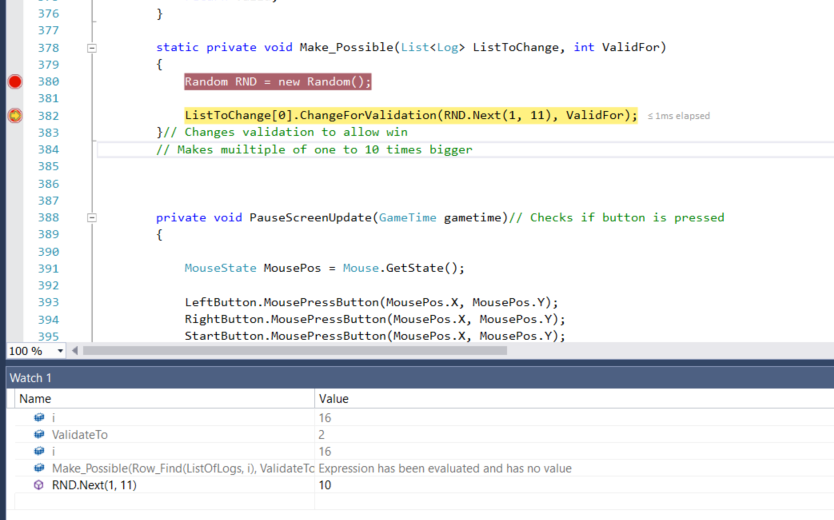
Log 2, Valid, current number = 14, 14 mod 2 (multiple selected) = 0

Returns valid row with possible solution doesn’t need to change first object in rows list to valid solution

Invalid solution

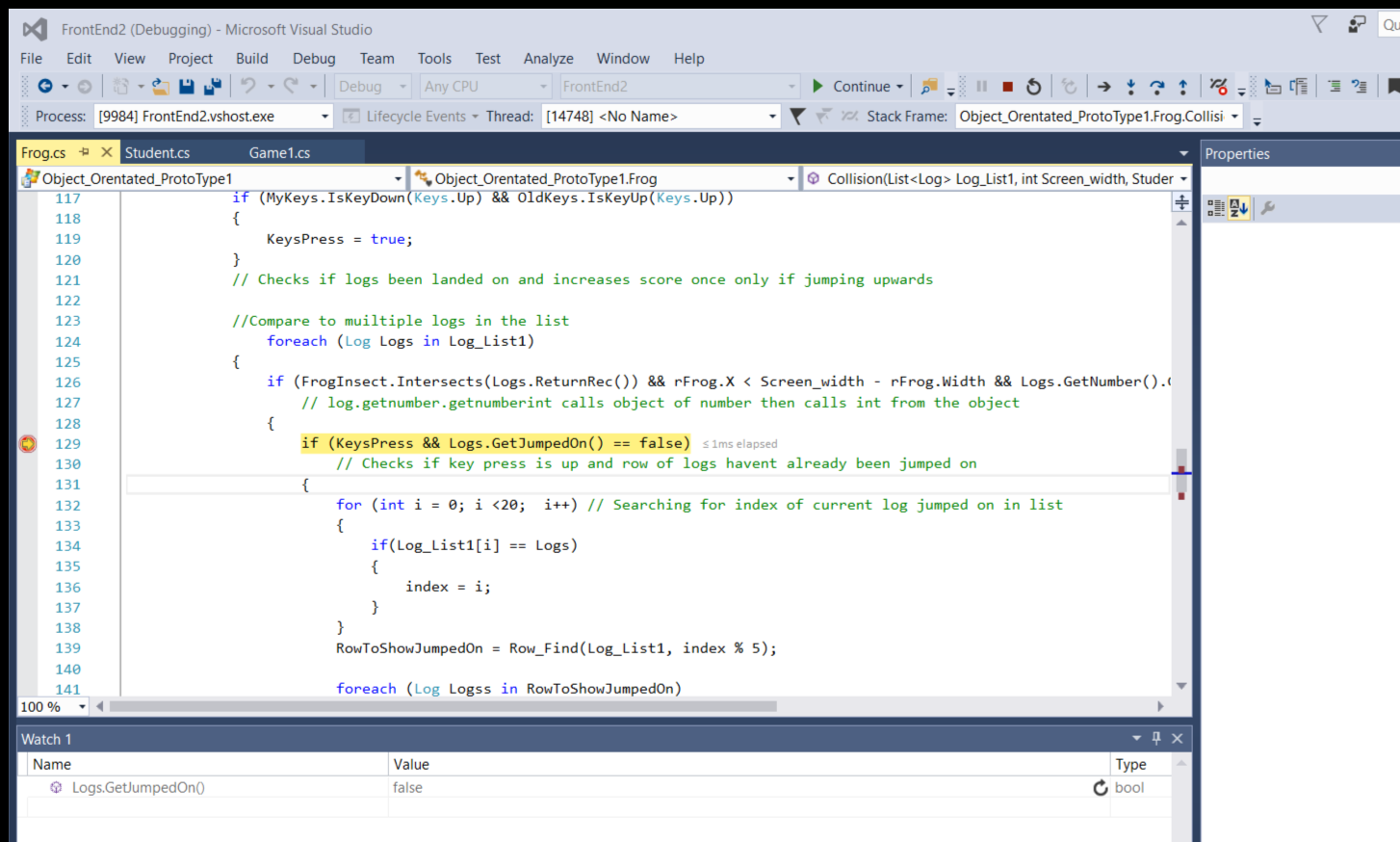


Current row 2 found no solutions

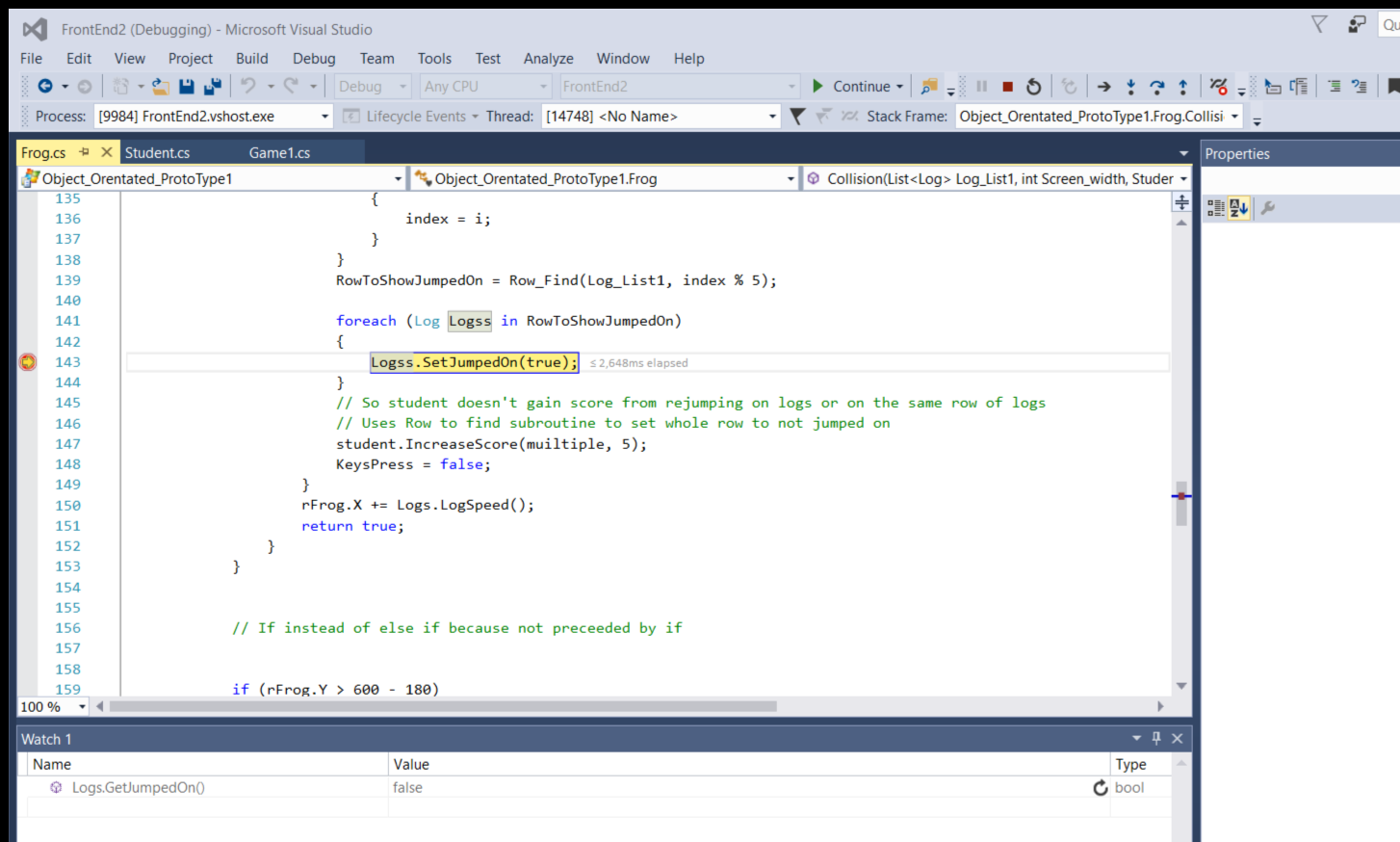


Changes first number of list to be a multiple of selected number (2) and random between 1 – 10 (10)

### Test 9

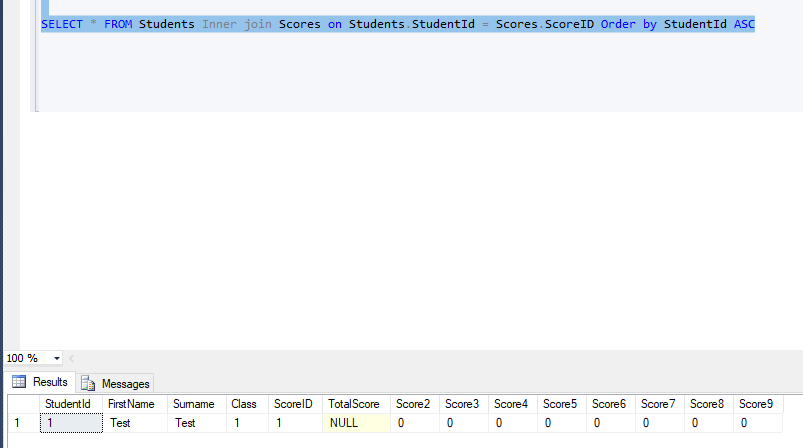


Checks if current log already jumped on currently false



Sets all logs on row to being jumped on

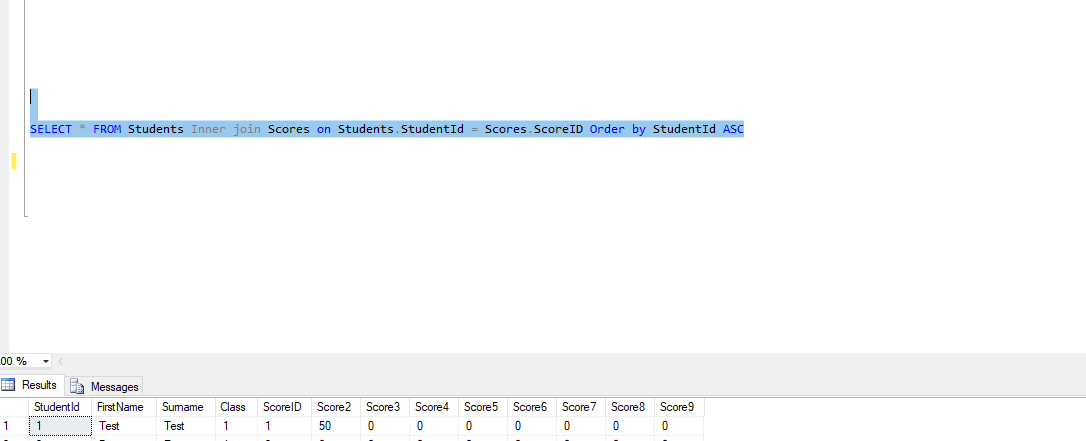
### Test 15



Score for multiple 2 = 0



Score for multiple 2 increased



Score changed