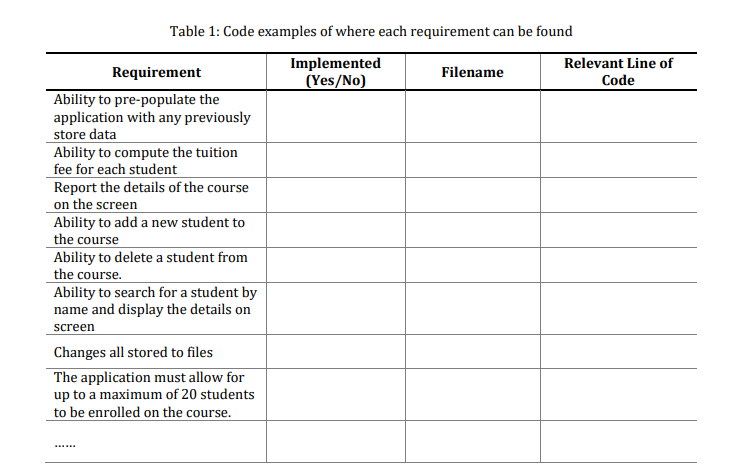
**Names: Jacob Mcilrath**

**Zakk Mcilrath**

**Roll no: B00884075**

**B00883173**



234

Course.java

Yes

Function at 206 course.java

17

57

187

73

55

89

162

Course.java

Course.java

Course.java

Course.java

Course.java

Student.java

Main.java

Yes

Yes

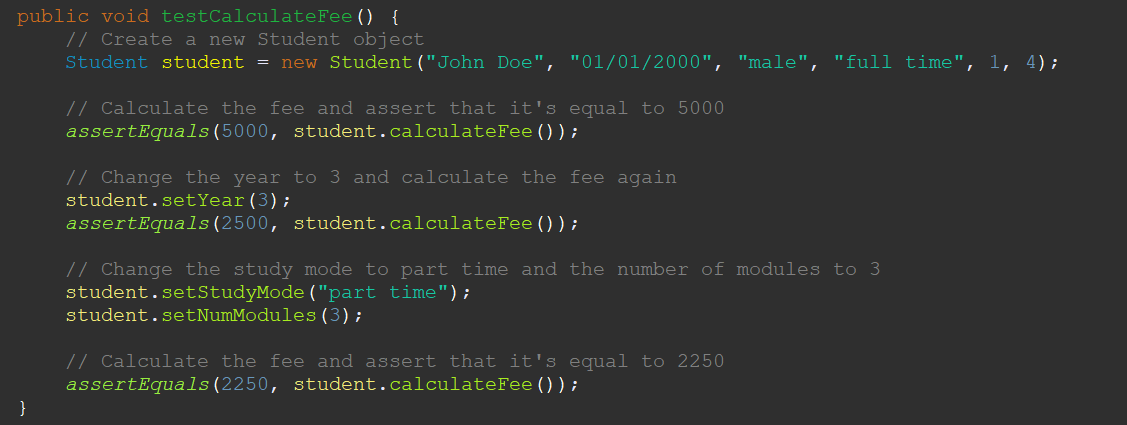
Yes

Yes

Yes

Yes

Yes

**Approach to test cases**  
Test case 1  


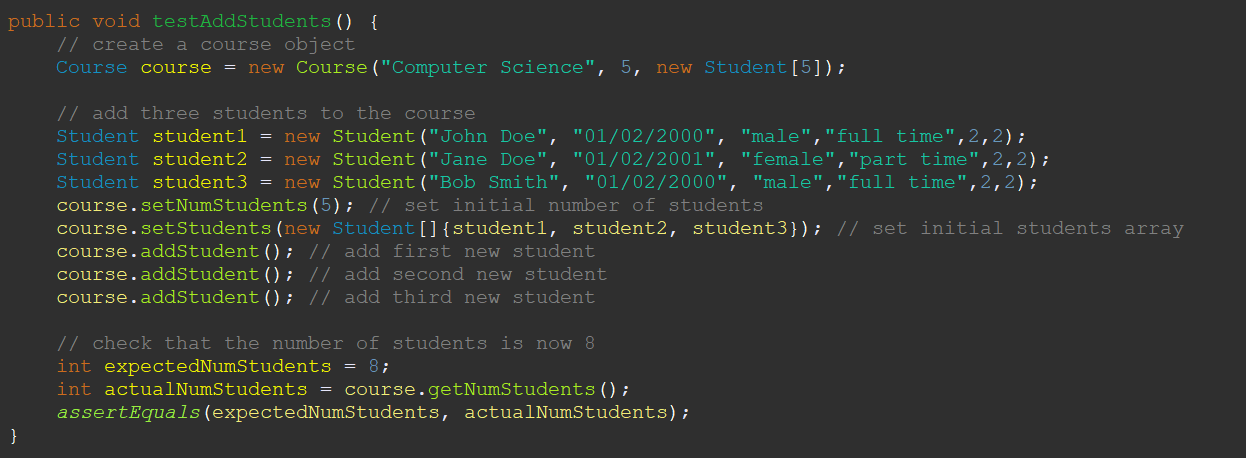
This test case is testing the **calculateFee()** method of the **Student** class, which calculates the fee that a student needs to pay based on their year of study, study mode (full-time or part-time), and the number of modules they are taking.

The test case creates a **Student** object with the name "John Doe", date of birth "01/01/2000", gender "male", study mode "full time", year of study 1, and number of modules 4. The **calculateFee()** method is then called on this object, and the expected result is that the fee should be 5000. This value is then checked using the **assertEquals()** method.

The test case then changes the year of study to 3 using the **setYear()** method, and calculates the fee again. This time, the expected fee is 2500 (since the fee is halved for students in their third year of study), and this value is checked using **assertEquals()**.

Finally, the test case changes the study mode to "part time" and the number of modules to 3, and calculates the fee again. This time, the expected fee is 2250 (since the fee is further reduced for part-time students), and this value is checked using **assertEquals()**.

Test case 2:



This test case is testing the **addStudent()** method of the **Course** class, which is responsible for adding a new student to a course.

The test case creates a **Course** object with the name "Computer Science", a maximum number of students of 5, and an empty array of students with a length of 5.

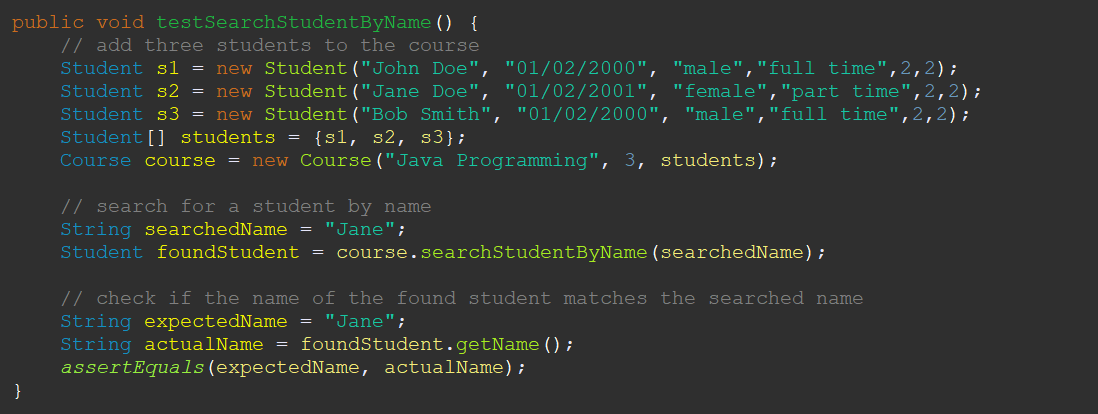
The test case then creates three **Student** objects, sets the initial number of students in the course to 5, and sets the initial students array to contain the first three students.

The **addStudent()** method is then called three times on the **Course** object, adding three new students to the course.

Finally, the test case checks that the number of students in the course is now 8 using **assertEquals()**.

Overall, this test case ensures that the **addStudent()** method of the **Course** class works correctly and that it can add new students to a course, increasing the number of students in the course accordingly.

Test case 3:



This test case is testing the **searchStudentByName()** method of the **Course** class, which searches for a student in a course by their name.

The test case creates three **Student** objects, and a **Course** object called "Java Programming" with a maximum number of students of 3 and the array of students containing the three students created.

The **searchStudentByName()** method is then called on the **Course** object, with the name "Jane" as the search parameter. The expected result is that the method should return the **Student** object with the name "Jane".

Finally, the test case checks whether the name of the found **Student** object matches the expected name "Jane", using **assertEquals()**.

Overall, this test case ensures that the **searchStudentByName()** method of the **Course** class works correctly and that it can find a **Student** object in a course by their name.