Author Robert Bonagura

Test cases for Outstate class.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Description | Input  (Method being tested, followed by the parameters it receives as input) | Expected Output  (The return value of each method tested in the previous column) |
| 1 | Checks default constructor and TuitionDue() based on students full time status, and toString() value of each | Case a: Full time  Constructor: (“Robert”, “Bonagura”, FULL\_TIME\_CREDIT\_MIN, false)  outstate1.TuitionDue():  outstate1.toString()  Case b: Part time  Constructor: (“Bob”, “Bonagura”, FULL\_TIME\_CREDIT\_MIN -1, false)  outstate2.TuitionDue():  outstate2.toString() | Case a:  outstate1  10513  Name: Robert Bonagura  Credits: 12  Lives in Trisate: false  Tuition Due: 10513  outstate2  9162  Name: Bob Bonagura  Credits: 11  Lives in Tristate: false  Tuition Due: 9162 |
| 2 | Tests TuitionDue() on full-time and part-time for both instate values | Case a: instate = true  Constructor: (“Robert”, “Bonagura”, FULL\_TIME\_CREDIT\_MIN, true)  outstate1.TuitionDue():  Case b: instate = false  Constructor: (“Greg”, “Bonagura”, FULL\_TIME\_CREDIT\_MIN , false)  outstate2.TuitionDue(): | Case a:  outstate1  8113  outstate2  10513 |
| 3 | Tests if student is taking a valid number of credits | Case a:  Constructor: (“Bobby”, “Bonagura”,  FULL\_TIME\_CREDIT\_MIN –  FULL\_TIME\_CREDIT\_MIN, false)  outstate1.isValid()  Case b:  Constructor: (“Bobby”, “Bonagura”,  FULL\_TIME\_CREDIT\_MIN, false)  outstate2.isValid() | Case a:  outstate1  false  Case b:  outstate2  true |

Test cases for International class.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Description | Input  (Method being tested, followed by the parameters it receives as input) | Expected Output  (The return value of each method tested in the previous column) |
| 1 | Checks default constructor and TuitionDue() based on students full time status, and toString() value of each | Case a: Full time  Constructor: (“Robert”, “Bonagura”, FULL\_TIME\_CREDIT\_MIN, false)  internat1.TuitionDue():  internat1.toString()  Case b: Part time  Constructor: (“Bob”, “Bonagura”, FULL\_TIME\_CREDIT\_MIN -1, false)  internat2.TuitionDue():  internat2.toString() | Case a:  internat1  13131  Name: Robert Bonagura  Credits: 12  Is exchange student: false  Tuition Due: 13131  internat2  11591  Name: Bob Bonagura  Credits: 11  IS exchange student: false  Tuition Due: 11591 |
| 2 | Tests TuitionDue() on full-time and part-time for both exchange values | Case a: exchange = true  Constructor: (“Robert”, “Bonagura”, FULL\_TIME\_CREDIT\_MIN, true)  internat1.TuitionDue():  Case b: exchange = false  Constructor: (“Greg”, “Bonagura”, FULL\_TIME\_CREDIT\_MIN , false)  internat2.TuitionDue(): | Case a:  internat1  1791  internat2  15966 |
| 3 | Tests if student is taking a valid number of credits | Case a:  Constructor: (“Bobby”, “Bonagura”,  FULL\_TIME\_CREDIT\_MIN –  FULL\_TIME\_CREDIT\_MIN, false)  internat1.isValid()  Case b:  Constructor: (“Bobby”, “Bonagura”,  FULL\_TIME\_CREDIT\_MIN, false)  internat2.isValid() | Case a:  internat1  false  Case b:  internat2  true |

Author Ezra Haleva

Test cases for Instate class.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Description | Sample Input | Expected result / output |
| #1 - 2 | Test constructor to ensure credit and funding are intitialized correctly. | An instance of Instate called instate1 has been initialized with  fName = “Ezra”  lName = “Haleva”  credit = 15  funding = 100 | #1)  instate1.credit equals 15  #2)  Instate1.funding equals 100 |
| #3 | Test isValid() to ensure it returns true for valid instances | An instance of Instate called validInstate has been initialized with  fName = “Ezra”  lName = “Haleva”  credit = 15  funding = 100 | validInstate.isValid() returns true |
| #4 | Test isValid() to ensure it returns false for invalid instances | An instance of Instate called invalidInstate has been initialized with  fName = “Ezra”  lName = “Haleva”  credit = 0  funding = 100 | validInstate.isValid() returns true |
| #5 | Test tuitionDue() returns credits\*COST\_PER\_CREDIT + FULL\_TIME\_FEE when credit is full time amount | An instance of Instate called instate2 has been initialized with  fName = “Ezra”  lName = “Haleva”  credit = 15  funding = 0 | Instate2.tuitionDue() returns cost per credit times 15 + full time fee |
| #6 | Test tuitionDue() returns MAX\_BILLABLE CREDITS\*COST\_PER\_CREDIT + FULL\_TIME\_FEE when credit is greater than MAX\_CREDITS | An instance of Instate called instate3 has been initialized with  fName = “Ezra”  lName = “Haleva”  credit = 18  funding = 0 | Instate2.tuitionDue() returns cost per credit times 15 + full time fee |
| #7 | Test tuitionDue() returns credits\*COST\_PER\_CREDIT + PART\_TIME\_FEE when credit is less than FULL\_TIME\_MIN | An instance of Instate called instate4 has been initialized with  fName = “Ezra”  lName = “Haleva”  credit = 11  funding = 0 | Instate2.tuitionDue() returns cost per credit times 11 + part time fee |
| #8 | Test tuitionDue() successfully adds funding to returned amount when credit is greater than full time minimum | An instance of Instate called instate5 has been initialized with  fName = “Ezra”  lName = “Haleva”  credit = 15  funding = 1000 | Instate2.tuitionDue() returns cost per credit times 15 + full time fee + 1000 |
| #9 | Test tuitionDue() does not add funding amount when credit is less than full time minimum | An instance of Instate called instate6 has been initialized with  fName = “Ezra”  lName = “Haleva”  credit = 11  funding = 1000 | Instate2.tuitionDue() returns cost per credit times 11 + part time fee |
| #10 | Test toString() is returning correct string representation of object | An instance of Instate called instate7 has been initialized with  fName = “Ezra”  lName = “Haleva”  credit = 11  funding = 1000 | instate7.toString() is equal to  “Name: Ezra Haleva\nCredits: 11\nFunding: 100” |