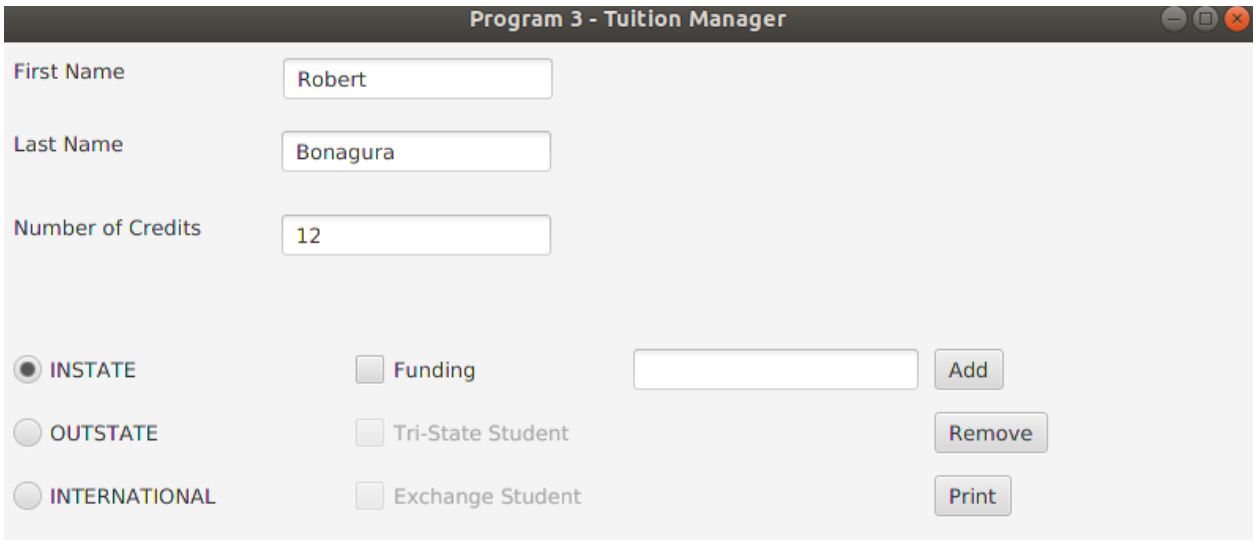
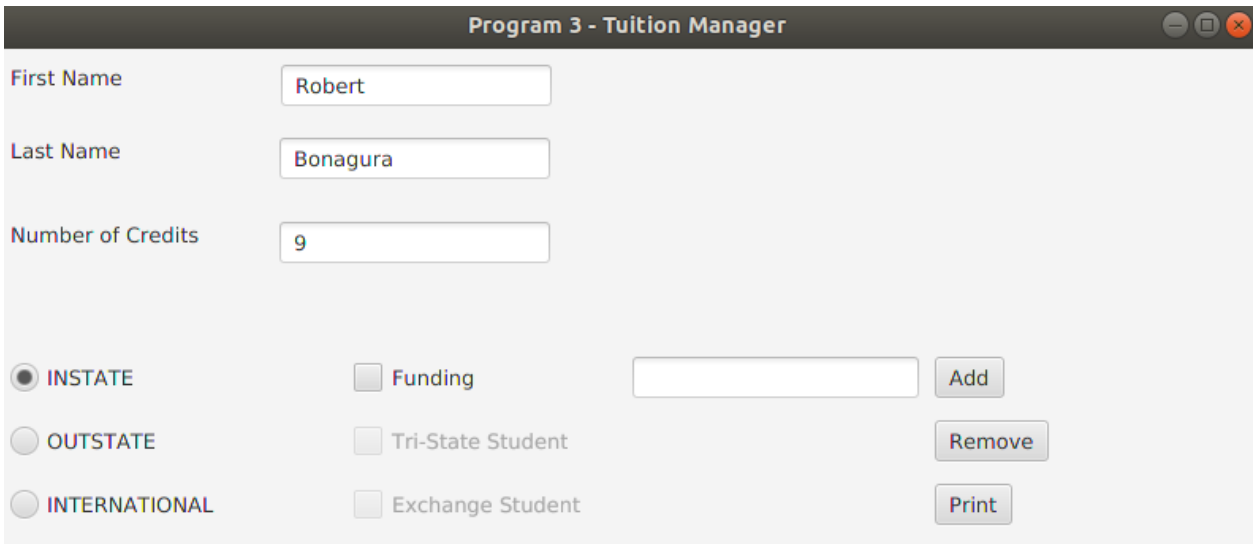
Test Doc

1. Tests TuitionDue() on both full-time and part-time students with no funding.
    a. Input



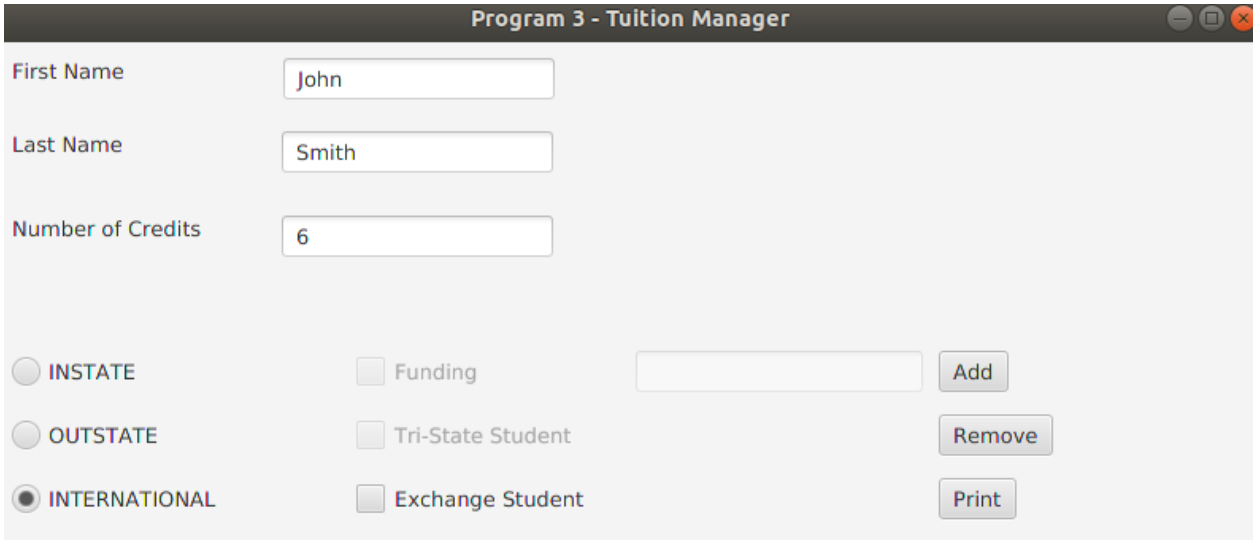
Tuition Due: 6637

    b. Input



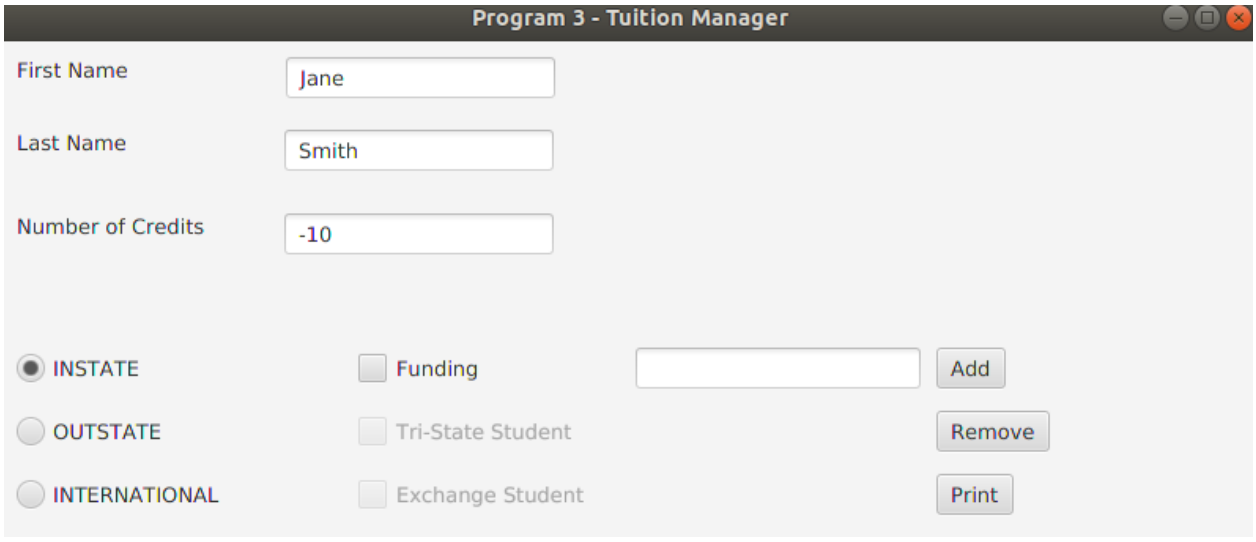
Tuition Due: 4743

2. Test if Student is taking a valid number of credits.
    a. Valid examples are shown in test case 1
    b. Invalid example Input



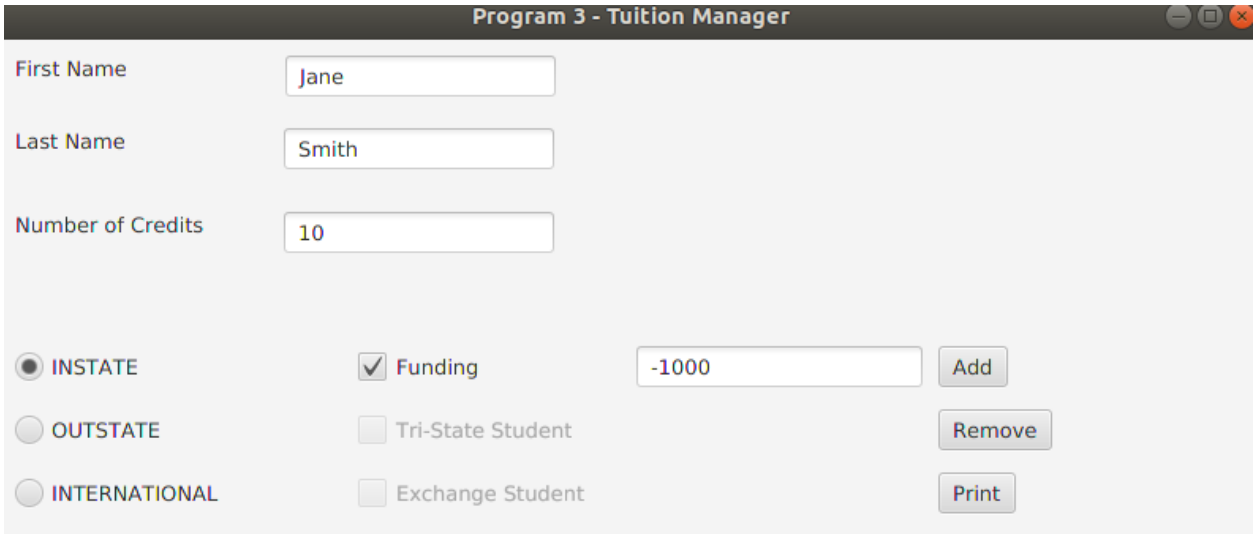
Output: “Error: International students must have at least 9 credits.”

3. 0 or negative value for number of credits or funding amount.
    a. Input



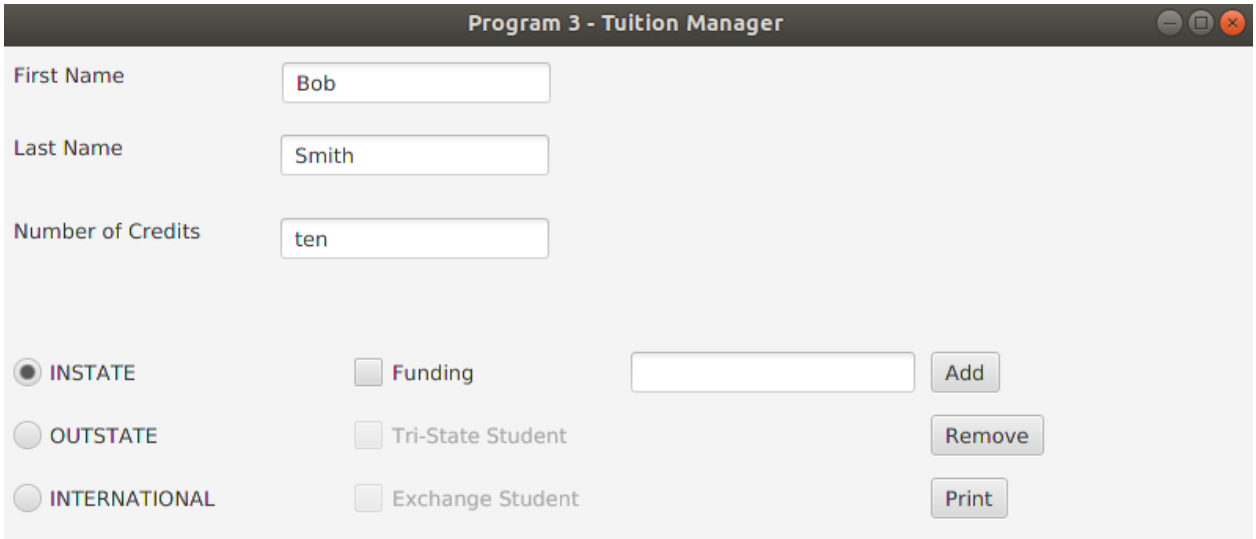
Output: “Error: Student must have a positive number of credits.”

b. Input



Output: “Error: Funding needs to be a positive value.”

4. Characters are entered where integers are expected
    a. Input



Output: “Error: Fields requiring a number value cannot be given a character value.”

5. Clicking ‘Add’ or ‘Remove’ buttons when no data has been entered.
    a. Output: “Error: Student name fields cannot be left blank or contain only white spaces.”

Additional Valid Test Case Examples

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)	Case a: outstate1 false Case b: outstate2

		outstate2.isValid()	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)	Case a: internat1 false Case b: internat2

		internat2.isValid()	true

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”