1. The most difficult part of this project was to write a series of if-else statements that give the correct output. I had a bug inputting student status. While the build of the program was successful, the program wouldn’t proceed even though I put the correct responses (y and n). The line “you must enter y or n” popped up no matter what I put there. I was sure it was a logic error since the program could run. Initially, I wrote “if (student != "y" || student != "n")” for outputting “you must enter y or n”. Then, I realized that both y and n always satisfy one of the conditions and would trigger the notice for invalid inputs. The correct way to write the condition is “if (student != "y" && student != "n")”, which requires the input to be different from both y and n. I fixed the logic error and then the program now proceeds successfully if valid responses are put. It was such a small error, but it made the whole program stop working as I expected.
2. 1. To ensure the program recognizes invalid inputs:

(Age of rider: -1 Result: The age must not be negative)

(Student? (y/n): i Result: You must enter y or n)

(Destination: (nothing) Result: You must enter a destination)

(Number of zone boundaries crossed: -1 Result: The number of zone boundaries crossed must not be negative)

All results came after the line of three hyphens.

2. To check if calculations are correct for normal cases and discounts:

Under 18: (16, y, Bay city, 0, $0.65)

18 and above (under 65), student: (18, y, Bay city, 1, $0.65)

18 and above (under 65), non-student: (25, n, Bay city, 5, $4.20)

65 and above, student: (65, y, Bay city, 0, $0.45)

65 and above, student: (65, y, Bay city, 1, $0.65)

65 and above, non-student: (65, n, Bay city, 0, $0.45)

65 and above, non-student: (65, n, Bay city, 1, $0.90)

All above ended with the line (under the line of three hyphens): The fare to *destination* is $*fare.*