**CS 31 Project 2 Report**

1. A notable obstacle I faced was having to account for the various conditions when calculating discounts. The fare for senior riders was especially notable as we had to calculate the cheapest fare available to them taking into account all the factors.
2. List of test data: Program handles all test data correctly  
   1. Rider is under 18 and crosses at most 1 boundary. (17, n, LA, 1) OR (17, y, LA, 1) 🡪 Discount condition  
   2. Rider is under 18 and crosses more than 1 boundary (17, n, LA, 3) OR (17, y, LA, 3) 🡪 No Discount condition  
   3. Rider is a student 18 or over and crosses at most 1 boundary (22, y, LA, 1) 🡪 Discount condition  
   4. Rider is 18 or over, is not a student and crosses any number of boundaries (22, n, LA, 3) 🡪 No Discount condition  
   5. Rider is a student 18 or over and crosses more than 1 boundary (22, y, LA, 3) 🡪 No Discount condition  
   6. Rider is a senior 65 or over and crosses 0 boundaries (68, n, LA, 0) OR (68, y, LA, 0) 🡪 Discount condition  
   7. Rider is a senior 65 or over, is not a student, and crosses 1 or more boundaries (68, n, LA, 3) 🡪 Discount condition  
   8. Rider is a senior student, 65 or over, and crosses at most 1 boundary (68, y, LA, 1) 🡪 Discount condition  
   9. Rider is a senior student, 65 or over and crosses more than 1 boundary (68, y, LA, 3) 🡪 Discount condition  
   10. Rider’s age is input as a negative number (-22, y, LA, 3) 🡪 Error test  
   11. Student status is input as something other than y or n (22, Y, LA, 3) OR (22, N, LA, 3) OR (22, yes, LA, 3) 🡪 Error test  
   12. If nothing was provided for the destination (22, y, , 3) 🡪 Error test  
   13. If number of zone boundaries entered is negative (22, y, LA, -3) 🡪 Error test