Report

1. Some of the obstacles I overcame were to compile a program that fits into different situations and can produce different outputs under each situation, and to input correct and accurate logics (such as which identifiers and flow controls to use) into the program so that it produces the correct output.

Also, I overcame the obstacle of correctly renaming the project files without prohibiting it to building or running.

1. Test cases:

* Put nothing for customer name to check if the program produces the output “--- You must enter a customer name.”
* Put negative numbers and 0 for miles driven and minutes driven to check if the program produces "--- The mileage must be positive."
* Put negative numbers and 0 for minutes driven and minutes driven to check if the program produces "--- The time must be positive."
* Put “BR-UBerX” for ride type to check if the program produces the output "--- The ride type is not valid."
* Put “Jingjing Nie” for customer name, 120 for miles driven and 130.0 for minutes driven, and “Br-UberX” for ride type to check if the program produces the correct output with name “Jingjing Nie” and values equal to the ones fare calculator gives ($129.15). (Similarly for “Br-UberXL” ride type)
* Put “Jingjing Nie” for customer name, 1.0 for miles driven and 1.0 for minutes driven, and “Br-UberX” for ride type to check if the program produces the correct output with name “Jingjing Nie” and values equal to the minimum fare ($5.15). (Similarly for “Br-UberXL” ride type)