Report

1. A notable discussion of obstacles you overcame:

In writing this project, I introduced quite a few “if statements”, due to the requirements of the spec. The program is designed to produce different outputs base on our inputs to the operating system and perform calculations based the input number. Whether the item is premium and in which range does the “units sent” fall create a great number of possible result. The hardest part is to figure out how to apply the appropriate equation to each case. Therefore, it is crucial to keep track of the opening and ending curly bracket. The habit of lining up the opening and ending bracket for the same statement turned out to be very useful in this project. It gave me a clear visualization of where does the statement begin and end.

Negative units sent (-20, “Run”, 10, y)

To test whether the program says, “The number of units sent must be nonnegative.”

Empty string (20, “”, 10, y)

To test whether the program says, “You must enter a title.”

Negative base price (20, “Run”, -10, y)

To test whether the program says, “The base price must be nonnegative.”

Neither yes nor no (20, “Run”, 10, w)

To test whether the program says, “You must enter y or n.”

Less than or equal to 400 units sent (20, “Run”, 10, y)

401 to 1200, premium item (800, “Run”, 10, y)

401 to 1200, non-premium item (800, “Run”, 10, n)

1201 and above, premium item (1500, “Run”, 10, y)

1201 and above, non-premium item (1500, “Run”, 10, n)

The five sets of data above are used to check whether the correct equations is used for the calculation