Before refactoring:

- In inside configuration_sim.cc which is in driver directory, the process of creating an instance of configuration_simulator called configSimulator, and updating simulation is duplicated from line 104 to 109 and from line 135 to 140.
- Inside bus.cc which is in src directory, the entire process of unloading passengers exists in Move method at line 131

After refactoring:

- RunningDefault method is created for creating an instance of configuration_simulator, start and update simulation. RunningDefault method resides at line 18 to 26 inside configuration_sim.cc which is in driver directory
- The process of unloading passengers now resides in UnloadPassengers method. UnloadPassengers method is defined from line 109 to 120. UnloadPassengers method is called at line 131.

In the file configuration_sim.cc that resides in driver directory, I modified the file to reduce the length of the main method by using Extract Method. It was clear that the code resides between line 105 to 110 and the code resides between line 136 to 141 are the same. Duplicating code inside a source file is commonly not a good coding practice. The reason is that if a bug is discovered in a block of duplicate code, the developer has to devote his resources to make changes in all places that the block of duplication code occurs. The more time spent on searching and maintaining duplicate code, the less time spent on debugging other bugs or adding valuable features to the product. In order to avoid such problem, refactoring by using Extract Method was used.

In the file bus.cc that resides in src directory, the loop that is used for unloading passengers from the bus is now moved from the Move method to the private method called UnloadPassenger. UnloadPassenger is called at line 131. By using Extract Method, this refactoring is to reduce the complexity of method Move by separating the process of unloading passenger and moving the bus. The separation hypothetically helps developers of this project to debug faster if the complexity of each method is kept as minimum as possible.