**Group Members**

Adrian Sterling

Daniell Harrow

Rheana Hagigal

Kimani King

Danielle Landeis

Carey O’Connor

Tresana Wright

The program creates a class, Hotel, that is used to store all the hotel related data required by the algorithm. Hotel, though named as such, actually represents a single room. For variable management, a vector of the class Hotel is created to organize suites.

Hotel contains the room number, the capacity, the rate of charge and status of a given room/suite. Virtualized functions for retrieving or modifying this information are used throughout the algorithm.

The first defined method, menu, has the sole purpose of making the graphical presentation of the menu for the user to refer to using cout, as is the same for all other output in the program.

The second defined method, addroom, accepts the vector of rooms as the parameter and prompts input to create a new room with the Hotel constructor, and pushes it to the end of the vector.

The third defined method, reserveroom, accepts the vector of rooms as the parameter. It verifies that there are more than three rooms in the system and before this method is allowed to be carried out. If the method is carried out, the room number is prompted and that room is selected via For loop. Once selected the information of the room will be displayed. If said room number was not found an error is raised.

The fourth defined method, modifyroom, accepts the vector of rooms as the parameter. The method will ensure there are rooms to modify in the first place, if there are indeed rooms ora room to modify then a room number is requested. A suite is selected with this number via For loop and Its information is displayed. A new guest amount and a new rate are prompted and then applied. If said room number was not found an error is raised.

The last method, viewroom, accepts the vector of rooms as the parameter and iterates it with a for loop, displaying all of the information in each.

These methods are integrated with the main code block, which displays the menu with "menu", executes a clear screen, accepts input for a selection and carries out one of the methods based on that selection via a Switch Case statement. The final selection, however, is different and is a cue for terminating the program.

**Testing and Debugging**

The program was tested using CodeBlocks, a popular C++ compiler that is free and reliable. The default CodeBlocks C++ compiler (GNU GCC) was used to compile this program. The program was built, run and executed with zero errors and four warnings and in an execution time of four seconds. The debugging process was done using CodeBlock’s pre-installed debugging feature and the proper configurations were selected. There were no bugs or defects presents during the debugging process and the program executed as expected.

The program was then tested with the intention of finding an error. Both white box and black box testing methods were utilized. During the white box testing process, the independent paths, logical branches, loops and data structures were tested and executed at least once to ensure their functionality.

During the black box testing process, the program was analyzed to locate any functional, interface and efficiency errors.