

## Hotel room reservation program

This documentation and the C++ code it refers to is provided in english to ensure combatability and internationality. The program consists of main and eight sub-programs/functions, which are explained below. The reservation data is save to the file reservationData.txt. If it doesn't already exist the user will be prompted to generate one.

### **int main()**

- Includes the main menu of the program.
- The menu has the options of reserving a new room (ReserveNew), searching for a reservation (SearchReserved) or exiting the program.
- The program will keep running until the command to exit is provided.

### **void NumberOfRooms()**

- The total number of hotel rooms is randomly generated using Mersenne Twister as defined by the range MIN\_ROOMS and MAX\_ROOMS (default 40-300).
- The number generated must be even and is split so that the first half of the rooms is for single bed rooms and the other half is for double bed rooms.

### **void ReserveNew()**

- Used for reserving a new room under a randomly generated unique reservation ID between 10000-99999 and the name of the reserver.
- Asks the user whether they want a single bed or a double bed room, asks for a valid room number and then prompts for a full name and the number of nights.
- Randomly chooses using Mersenne the discount factor of either 0%, 10% or 20% for each room reserved.
- Saves the reservation ID, room number and full name to the arrays RoomsID[], RoomsNumber[] and RoomsName[] with the room number as the index.
- The user can reserve as many rooms as they wish up to the number of available rooms.

### **void SearchReserved()**

- Allows the user to search for pre-existing reservations using a corresponding reserver name, reservation ID or room number.
- If multiple matches with the same name come up then prompt the user to input the relevant reservation ID or room number.
- When a match is found prompt the user to remove the reservation.

### **bool ReservationStatus()**

- A simple function to check if the provided room number is reserved by checking against the corresponding index of RoomsID[] for the existence of a non-zero value. Returns true if the room is reserved.

### **fileExists()**

- Check if the files containing the reservation data exist.
- Check if the file is empty.

### **bool IsValidFullname()**

- A comprehensive function that validates and modifies a given full name. The help of ChatGPT was used for implementing the functionality.
- Trims leading and trailing spaces at the start and end of the name.
- Makes sure that the name contains only alphabetic characters and spaces and no digits or special characters except hyphens '-' for middle names.
- The first letter of a name must be uppercase.
- Returns true if the name is valid. Otherwise false.

### **string MenuChoice()**

- Used for streamlining user input when there are two options to choose from i.e. 1 and 2. Validates the user input and returns a string that can be a "1", "1.", "2" or "2.".

### **int ChooseRoom()**

- After the user chooses to reserve either a single bed room or a double bed room, run this function.
- Gives the user the option to either manually or automatically choose the room number. Depending if the user wants a single bed room or a double bed room, give range numbers representing the range of rooms for single or double bed rooms as arguments (two ints)