Hotel Reservation Program

made by Roman Zinkevich

This hotel reservation program has all required functionalities.

1. Booking. User can book a room in this program. He can choose what type of room to book, for how many nights and needs to type his name. User will be asked ones if his reservation information is correct.
   a) He can change it if not
   b) He can cancel that reservation
   c) If everything is correct program goes futher. User will get reservation number that would be necessary if he wants to delete reservation. Or if there is no more available rooms program will go to the main menu.
2. Booking view. User can check information of the room he booked. For that he needs to type reservation number (given after reservation) or his name. After that user will see all information about his reservation if it exists (price, number of nights, type of room, name of guest, reservation number), if not program will tell that reservation doesn't exist. User can't check information if no reservations is booked.
3. Removing a reservation. If user wants to cancel his reservation, he can do it in that program. He needs to type number of reservation he wants to delete (not name of the guest because user might have multiple reservations). User can't delete reservation if no reservations is booked.
4. Quit. User can quit the program.

   I believe my program deserves 5 because all requirements are met and bugs are not found.

Code of the program:

```
#include <string>

//import of "string" that helps with getline()

#include <iostream>

//import of cin,cout

#include <ctime>

//import of time module

using namespace std;

//import all std methods e.g cin,cout



//structure of 1 reservation, that contains type of room (1-single room,2-double room),

//number of reservation(randomly generated), number of nights, cost of the reservation,

//name of the person who reserved

struct Reservation {

	short int room_type = 0;

	int number;

	int nights=1;

	double cost=1;

	string name;
};



//function that checks if input value contains only of numbers
bool check_integer(const string S) {
    //goes through all elements of string S from S[0] to S[last]
    for (int i = 0; i < S.length(); i++) {
        //checks if element of string S between 0 and 9, that is
element is a number
        //more than that checks if S contains more than 5
elements
        //for that program you can't enter numbers bigger than
99999 or smaller than -9999, because program requires only smaller
numbers.
        if ((S[i] < '0') || (S[i] > '9')||(S.length()>5)) {
            //if all above mentioned is not fine, then input is
incorrect
            cout << "Incorrect input"<<endl<<endl;
            return false;
        }
    }
    return true;
}
```

```
//function that checks if number input is correct according to different for every usage requirements
int int_check_input(const int MINIMUM, const int MAXIMUM,const string EXCEPTION_TEXT_MINIMUM, const string EXCEPTION_TEXT_MAXIMUM, const string TEXT_BEFORE_INPUT) {
	string answerstring;
	int answer;
	//checkpoint for input if something went wrong and program needs new input
	input:
	//prints text that explains what you need to print
	cout << TEXT_BEFORE_INPUT<<endl;
	//takes what user types
	getline(cin, answerstring);
	//if length of the input is 0, that is ENTER, input is incorrect
	if (answerstring.length()==0) {
		cin.clear();
		cout << "Incorrect input" << endl << endl;
		//goes back to checkpoint input
		goto input;
	}
	//if check_integer function thinks that input is not integer or bigger than 99999 or smaller than -9999
	//then program goes back to checkpoint input
	if (check_integer(answerstring) == false)
```

```
        goto input;
    //transfer input value to integer variable
    answer = stoi(answerstring);
    //if input value is bigger then required value
    if (answer > MAXIMUM) {
        //prints exception that explains why input is incorrect
        cout << EXCEPTION_TEXT_MAXIMUM << endl;
        goto input;
    }
    //if input value is smaller then required value
    else if (answer < MINIMUM){
        //prints exception that explains why input is incorrect
        cout << EXCEPTION_TEXT_MINIMUM << endl;
        goto input;
    }
    //if above mentioned is fine then function can return input value
    else {
        return answer;
    }

}

//function that checks is string input is correct
```

```
string string_check_input(const string& TEXT_BEFORE_INPUT){
    string answer;
    //checkpoint for input if something went wrong and program needs new input
    input:
    //prints text that explains what you need to print
    cout << TEXT_BEFORE_INPUT << endl;
    //takes what user types
    getline(cin, answer);
    //if length of the input is 0, that is ENTER, input is incorrect
    if ((answer.length() == 0)||(answer==" ")) {
        cin.clear();
        cout << "Incorrect input" << endl << endl;
        //goes back to checkpoint input
        goto input;
    }
    //if program comes here than everything is fine with input and it can return it
    return answer;
}

//function that creates random number of reservation between 10000 and 99999
int create_number(Reservation reservations[80]) {
	srand(time(0));//creates random number according to seed that contains current data
	int res_number = rand() % 90000 + 10000;//makes random number to be between 10000 and 99999
	//goes from 0 to 79
	for (int i = 0; i < 80; i++) {
		//while created before random number is already contained in some element of array of reservations
		//than program needs to create new number
		while (reservations[i].number==res_number)
			res_number=rand() % 90000 + 10000;
	}
	//if created number is new for array then program returns it
	return res_number;
}

//function that randomly creates discount
int create_discount() {
	srand(time(0));//creates random number according to seed that contains current data
	return rand() % 3 * 10;//makes number to be either 0 or 10 or 20
}

//function that changes information of reservation
```

```
void change_info(Reservation& reserv) {
	int answer = 1;//initially input of user is 1, so while loop can work
	//if input is 4 then user don't want to change anything and while loop can stop
	while (answer != 4) {
		//takes input from user through int_check_input function
		answer = int_check_input(1, 4, "Sorry, I don't understand", "Sorry, I don't understand", "\nWhat do you want to change?\n1.Type of room\n2.Amount of nights\n3.Name\n4.I don't want to change anything\n");
		switch (answer) {
		case 1://if user chooses 1(change the type of room)
			//takes input from user through the int_check_input function and changes type of room in reservation
			reserv.room_type = int_check_input(1, 2, "Sorry, I don't understand", "Sorry, I don't understand", "1.Single room\n2.Double room\n");
			break;
		case 2:
			//takes input from user through the int_check_input function and changes number of nights in reservation
			reserv.nights = int_check_input(1, 365, "You can't book less than 1 night", "365 is maximum, sorry", "How many nights?(365 is maximum)\n");
			break;
		case 3:
```

```
            //takes input from user through the int_check_input function and changes name of user in reservation
            reserv.name = string_check_input("Tell your name please");
            break;
        case 4:
            //ends the while loop
            cout << endl;
        }
    }
}

//function that creates reservation and takes all the info from user
void book_room(Reservation reservations[80],int& i, int& single_rooms, int& double_rooms) {
    double discount;
    //checks availability of single and double rooms, because user shouldn't be able to book unavailable room
    bool available_single= single_rooms != 0, available_double= double_rooms != 0;
    string text, text_single_room, text_double_room,room_type;
    //final option is initially 2, so while loop can be started
    int final_option=2;
    //checkpoint for input of type of room
    room_type_input:
```

```
    if (available_single)//checks if there are any single rooms
        //if so saves text about booking a room in a variable
        text_single_room = "Press 1 for single room (100 euro/night) (" + to_string(single_rooms) + " left)\n";
    else
        //if not save text about no single rooms in a variable
        text_single_room = "Sorry, there is no available single rooms\n";
    if (available_double)//checks if there are any doublee rooms
        //if so saves text about booking a room in a variable
        text_double_room="Press 2 for double room (150 euro/night) (" + to_string(double_rooms) + " left)\n";
    else
        //if not save text about no single rooms in a variable
        text_double_room="Sorry, there is no available double rooms\n";
    //if there is no single or double rooms tells user about that and goes back to main menu
    if (!available_double&&!available_single) {
        cout << "Sorry, there is no available rooms" << endl << endl;
        //makes room_type equal 4
        //so in main function it can make right decision to what type of room amount decrease in main function
        reservations[i].room_type = 4;
        return;
    }


    //creates whole introduction to reservation text
    text = "What kind of room do you want?\n"+text_single_room+text_double_room+ "Press 3 to go to the main menu";
    //takes input from user through int_check_input function and put it in reservation structure as type of room
    reservations[i].room_type = int_check_input(1, 3, "I don't understand", "I don't understand", text);
    //if there is no any kind of rooms and user chose that type tells user about that
    if (((single_rooms == 0) && (reservations[i].room_type == 1)) || ((double_rooms == 0) && (reservations[i].room_type == 2))){
        cout << "There is no available rooms" << endl;
        //goes back to checkpoint room_type_input so user can give new value
        goto room_type_input;
    }
    //if user chooses '3' then program goes back to main menu
    if (reservations[i].room_type == 3) {
        //makes room_type equal 4
        //so in main function it can make right decision to what type of room amount decrease in main function
        reservations[i].room_type = 4;
        return;
    }
    //takes input from user through int_check_input function and put it in reservation structure as number of nights
    reservations[i].nights = int_check_input(1, 365, "You can't book 0 nights", "365 is maximum, sorry", "How many nights? (365 is maximum)");

    //takes input from user through string_check_input function and put it in reservation structure as name of user
    reservations[i].name=string_check_input("Tell your name please");
    //creates random number of reservation with create_number function and put it in reservation structure as number of reservation
    reservations[i].number = create_number(reservations);
    //ends loop if final_option equals 2
    while (final_option == 2) {
        //initially cost of the reservation is 100 (per night)
        reservations[i].cost = 100;
        room_type = "single";//initially string variable room_type is single
        if (reservations[i].room_type == 2){//if chosen reservations room_type equals 2 (double room),
            //then string variable room_type is double
            room_type = "double";
            //and reservations cost increases by 50 (per night)
            reservations[i].cost += 50;
```

```
		}
		//multiplies cost of the reservation per night and amount of the nights
		reservations[i].cost *= reservations[i].nights;

		//randomly creates discount using function create_discount()
		discount = create_discount();
		reservations[i].cost *= (100-discount) / 100;//changes cost of the reservation according to discoun
		//prints all information about reservation
		cout << endl<<"All information about your reservation:" << endl;
		cout << "You have booked " << room_type << " room for " << reservations[i].nights << " nights" << endl;
		cout<<"It will cost you "<< reservations[i].cost<<" euros with "<<discount<<"% discount"<<endl;
		cout << "Your name is " << reservations[i].name << ", your reservation number is " << reservations[i].number << endl << endl;

		//asks if the reservation's info is correct
		final_option = int_check_input(1, 3, "Sorry, I don't understand", "Sorry, I don't understand", "Is your reservation info correct?\n1.Yes\n2.No, I want to change something\n3.No, I want to cancel my order");
		switch (final_option) {
		case 1://if correct than program can finish it
```

```
            cout << "Your reservation is finished" << endl<<endl;
            return;
        case 2://if not it goes to function change_info
            change_info(reservations[i]);
            break;
        case 3://if user wants to delete it than program will put
default values in the element of array and finish reservation
            cout << endl;
            reservations[i].cost = 1;
            reservations[i].nights = 1;
            reservations[i].name = "";
            reservations[i].number = -858993460;
            reservations[i].room_type = 4;
            return;
        }
    }
}

//function that looks for reserved room and prints its information if it
exists
void check_room(const Reservation RESERVATIONS[80]) {
    string name, room_type;
    //takes input from user through string_check_input function
and put it in the variable
```

```
	name=string_check_input("Tell me your name or reservation number to find your reservations");
	room_type = "single";//by default room_type is single but if it isn't it will be changed in the future
	cout << "Searching for information about your reservation(s):" << endl<<endl;
	for (int i = 0; i < 80; i++)/*goes through 0 to 79 in order to look through all elements of RESERVATIONS array */ {
		//if input data equals name or number in some reservation and that reservation is not filled with default value(cost=1)
		//then required reservation is found
		if ((RESERVATIONS[i].name == name)||(to_string(RESERVATIONS[i].number)==name)&&(RESERVATIONS[i].cost!=1)) {
			if (RESERVATIONS[i].room_type == 2) //if type of room is double than program needs to change room_type variable
				room_type = "double";
			//prints all useful information about reservation
			cout << "You have booked " << room_type << " room for " << RESERVATIONS[i].nights << " night(s)" << endl;
			cout << "It will cost you " << RESERVATIONS[i].cost << " euros" << endl;
			cout << "Your name is " << RESERVATIONS[i].name << ", your reservation number is " << RESERVATIONS[i].number << endl << endl;
			return;//goes back to main menu because required reservation is found
		}
	}
	//if reservation wasn't found than program goes here and it needs to print appropriate text
	cout << "Sorry, no booked rooms" << endl << endl;
}




//function that fixes order of elements in array after deleting one of the reservations
int fix_array(Reservation reservations[80], int deleted_index) {
	int i = deleted_index;
	if (i != 79)//if index of deleted element is not last
		//while next element of array is not empty program puts information from next element to the current element
		//that way all elements in the array will be without any spaces between them
		while ((reservations[i + 1].number != -858993460) && (reservations[i + 1].cost != 1)) {
			reservations[i].number = reservations[i + 1].number;
			reservations[i].cost = reservations[i + 1].cost;
			reservations[i].nights = reservations[i + 1].nights;
			reservations[i].name = reservations[i + 1].name;
```

```
            reservations[i].room_type = reservations[i + 1].room_type;
            i++;
        }
    //makes last element before empty one empty, because it was already copied by previous element
    reservations[i].cost = 1;
    reservations[i].nights = 1;
    reservations[i].name = "";
    reservations[i].number = -858993460;
    reservations[i].room_type = 4;
    return i;//return index of the first empty element so new reservation can be put after last one
}

//function that deletes required reservation
void delete_reservation(Reservation reservations[80],int& index,int& single_rooms,int& double_rooms) {
    int answer,number;
    enter_number://checkpoint if input was incorrect
    //takes input from user through int_check_input function and put it in a variable
    answer= int_check_input(1, 99999, "Sorry, wrong input", "Sorry, wrong input", "Enter reservation number to continue or\n1.To return to main menu");
    switch(answer) {
```

```
    case 1://if user enters 1 then program goes back to main menu
        return;
        break;
    default://if not program tries to find reservation number user
put
        for (int i = 0; i < 80; i++)/*goes through 0 to 79 in order to
look through all elements of RESERVATIONS array */ {
            //if one of the reservation's number equals input
data then program needs to delete it
            if (reservations[i].number == answer) {
                if (reservations[i].room_type == 1)//if the
reservation had single room
                    //then amount of available single rooms
needs to be increased
                    single_rooms++;
                else
                    //if it had double room than amount of
available double roomd needs to be increased
                    double_rooms++;
                //goes to fix_array function to make array
looks better
                //index of the first empty element goes to
'index' variable
                index=fix_array(reservations, i);
                cout << "You don't have that reservation
anymore\n\n";
```

```
                return;//goes back to main menu
            }
        }
        //if reservation wasn't found than program goes here and it needs to print appropriate text
        cout << "We can't find that reservation";
        //goes back to checkpoint to take new input data
        goto enter_number;
    }
}

int main() {
    srand(time(0));//initializes random number
    int num = 1; //initially choice of the user is 1
    int ROOMS_AMOUNT = (rand() % 21 + 20) * 2;//creates random even amount of rooms between 40 and 80
    int single_rooms = ROOMS_AMOUNT / 2, double_rooms = ROOMS_AMOUNT / 2;//amount of single and double room is half of all rooms
    int i = 0;//initially first element of the array has index 0
    Reservation reservations[80];//creates array of empty reservations
    while (num != 4) {//while user doesn't type 4 program will work
        //takes input from user through int_check_input function and put it in a variable
        num = int_check_input(1, 4, "Sorry, I don't understand", "Sorry, I don't understand", "What do you want to do?\n1.Book a room\n2.Check reservation\n3.Delete reservation\n4.Quit");
        cout << endl;
        switch (num) {
        case 1://if input datum is 1 then user wants to book a room
            //goes to booking function
            book_room(reservations,i, single_rooms, double_rooms);
            if (reservations[i].room_type == 4) {//if reservation's type is '4' then reservation was cancelled inside booking function
                reservations[i].room_type = 0;//makes default room type in that (cancelled) reservation
                //makes index smaller by 1 because reservation is cancelled
                //and future reservations need to be put in the array without empty reservations between each other
                i--;
            }
            else if (reservations[i].room_type == 1)//if reservation's type is '1' then it is single room
                single_rooms--;//lowers amount of single rooms
            else // if none of that then reservation's type is '2' which means double room
```

```
                double_rooms--;//lowers amount of double
rooms
            i++;//increases index of the array so next
reservations won't cover current one
            break;
        case 2://if input datum is 2 then user wants to see
information about already booked room
            if (i == 0) {//if index of array is 1 then there is no
reservations and nothing can be found
                cout << "There is no reservations yet\n";
                break;
            }
            //if index is not 0 then goes to function that prints
information about booked reservations
            check_room(reservations);
            break;
        case 3://if input datum is 3 then user wants to delete
previous reservation
            if (i == 0) {//if index of array is 1 then there is no
reservations and nothing can be deleted
                cout << "There is no reservations yet\n";
                break;
            }
            //if index is not 0 then goes to function that delete
reservation
```

```
    delete_reservation(reservations,i,single_rooms,double_rooms);
            break;
        case 4://if input datum is 4 then user wants to quit the
programm
            return 0;
        default://if input datum is anything else then it is wrong
and program needs to tell user that he typed incorrect answer
            cout << "Sorry, I don't understand"<<endl;
        }
    }
}
```