```csharp
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace assignment_2
12 {
13     public partial class Form1 : Form
14     {
15         const int ROOMS_PER_FLOOR = 5;
16         const int NUMBER_OF_FLOORS = 7;
17
18         private BindingList<HotelRoom> _hotelRooms;
19         Random rand = new Random();
20
21         public Form1()
22         {
23             InitializeComponent();
24             _hotelRooms = new BindingList<HotelRoom>();
25             listBox1.DataSource = _hotelRooms;
26
27
28             comboBoxRoomType.DataSource = Enum.GetValues(typeof(RoomType));
29
30             for (int i = 1; i <= ROOMS_PER_FLOOR*NUMBER_OF_FLOORS; i++)
31             {
32                 HotelRoom room = new HotelRoom((((i-1)/ROOMS_PER_FLOOR)*100)
                     +((i-1)%ROOMS_PER_FLOOR),((i-1)/ROOMS_PER_FLOOR),(RoomType)
                     rand.Next(0,5),rand.Next(0,2)>0,rand.Next(1,100),rand.Next
                     (0,2)>0);
33                 _hotelRooms.Add(room);
34             }
35             updateStatus();
36
37
38         }
39
40         private void buttonAdd_Click(object sender, EventArgs e)
41         {
42             int roomNumber = (int)numericUpDownRoomNumber.Value;
43             // floorNumber = (int)numericUpDownFloorNumber.Value;
44             RoomType roomType = (RoomType)comboBoxRoomType.SelectedItem;
45             bool riverView = checkBoxRiverView.Checked;
46             bool booked = checkBoxBooked.Checked;
47             decimal rate = numericUpDownRate.Value;
48
49             foreach(HotelRoom hotelRoom in _hotelRooms)
50             {
51                 if (hotelRoom.roomNumber == roomNumber)
52                 {
53                     return;
```

```csharp
54                     }
55                 }
56
57                 HotelRoom room = new HotelRoom                                    ⏎
                     (roomNumber,roomNumber/100,roomType,riverView,rate,false);
58                 _hotelRooms.Add(room);
59                 updateStatus();
60             }
61         private void buttonEdit_Click(object sender, EventArgs e)
62         {
63             HotelRoom room = new HotelRoom(_hotelRooms                            ⏎
                 [listBox1.SelectedIndex].roomNumber, (int)                          ⏎
                 numericUpDownRoomNumber.Value/100, (RoomType)                       ⏎
                 comboBoxRoomType.SelectedItem, checkBoxRiverView.Checked,           ⏎
                 numericUpDownRate.Value, checkBoxBooked.Checked);
64             _hotelRooms[listBox1.SelectedIndex] = room;
65             updateStatus();
66         }
67
68         private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
69         {
70             if (listBox1.SelectedIndex != -1)
71             {
72             HotelRoom room = (HotelRoom) listBox1.SelectedItem;
73             numericUpDownRoomNumber.Value = room.roomNumber;
74             numericUpDownFloorNumber.Value = room.floor;
75             comboBoxRoomType.SelectedItem = room.typeOfRoom;
76             checkBoxRiverView.Checked = room.riverView;
77             checkBoxBooked.Checked = room.booked;
78             numericUpDownRate.Value = room.rate;
79             }
80         }
81
82
83         private void numericUpDownRoomNumber_ValueChanged(object sender,          ⏎
             EventArgs e)
84         {
85             numericUpDownFloorNumber.Value = (int)                                ⏎
                 numericUpDownRoomNumber.Value / 100;
86         }
87
88         private void buttonDelete_Click(object sender, EventArgs e)
89         {
90             List<int> listSelected = new List<int>();
91             foreach (int index in listBox1.SelectedIndices)
92             {
93                 listSelected.Add(index);
94             }
95             listSelected.Sort();
96             for (int i = listSelected.Count - 1; i >= 0; i--)
97             {
98                 int index = listSelected[i];
99                 _hotelRooms.RemoveAt(index);
100             }
101             listBox1.ClearSelected();
102             updateStatus();
```

```csharp
103            }
104
105        private void buttonBook_Click(object sender, EventArgs e)
106        {
107            decimal invoice = 0;
108            bool roomsFree = false;
109            List<int> listSelected = new List<int>();
110            foreach (int index in listBox1.SelectedIndices)
111            {
112                if (_hotelRooms[index].booked)
113                {
114                    roomsFree = true;
115                }
116            }
117            if (!roomsFree)
118            {
119                foreach (int index in listBox1.SelectedIndices)
120                {
121                    /*doesn't work cannot figure out why. will randomly skip
                       entries
122                    HotelRoom room = _hotelRooms[index];
123                    room.booked = true;
124                    _hotelRooms[index] = room;
125                    */
126                    _hotelRooms[index].booked = true;
127                    invoice += _hotelRooms[index].rate;
128
129                }
130                listBox1.DataSource = null;
131                listBox1.DataSource = _hotelRooms;
132                updateStatus();
133                MessageBox.Show("all rooms booked with a total cost of " +
                    invoice.ToString("c"));
134            }
135            else
136            {
137                MessageBox.Show("1 or more of the rooms you selected are
                    already booked");
138            }
139
140        }
141        private void updateStatus()
142        {
143            int booked = 0;
144            int vacant = 0;
145            bool noVacancy = false;
146            foreach (HotelRoom hotelRoom in _hotelRooms)
147            {
148                if (hotelRoom.booked)
149                {
150                    booked++;
151                }
152                else
153                {
154                    vacant++;
155                    noVacancy = true;
```

```
156                    }
157                }
158            labelStatus.Text = "There are " + booked.ToString() + " rooms      ⏎
                  booked and " + vacant.ToString() + " rooms vacant";
159            if (noVacancy)
160            {
161                labelNoVacancy.BackColor = Color.Black;
162            }
163            else
164            {
165                labelNoVacancy.BackColor = Color.Red;
166            }
167        }
168    }
169 }
170
```