# CS145 – PROGRAMMING ASSIGNMENT 18

RESORT HOTEL

## OVERVIEW

This assignment will give you practice with object, arrays and strings

You are going to write a program that will maintain an active use of a number of different rooms at a resort, where each room is part of a particular building.

## BACKGROUND

A modern resort tends to not have a single building with rentable rooms, but rather a series of building, each with its own set of rooms. We want to be able to write a program to maintain an active directory of which rooms are occupied currently, which are empty, be able to rent out a room, and be able to state when the occupants have checked out of a room.

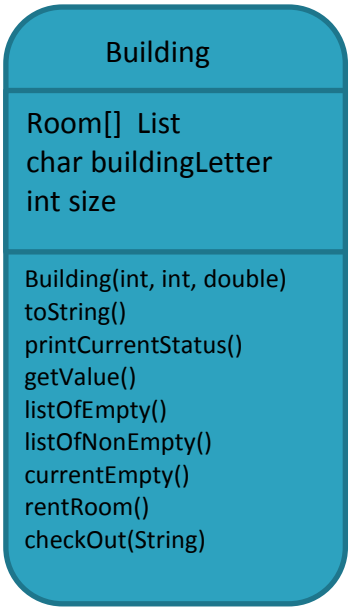
For the purposes of this assignment, we are going to assume that each building is given a character name and a number of rooms, and a price per room. Each room of that building will be assumed to be the same price. So for example Building 'A' might have 5 rooms each at a price of $150.00 a night. While building 'D' might have 10 rooms at a price of $75.00 a night.

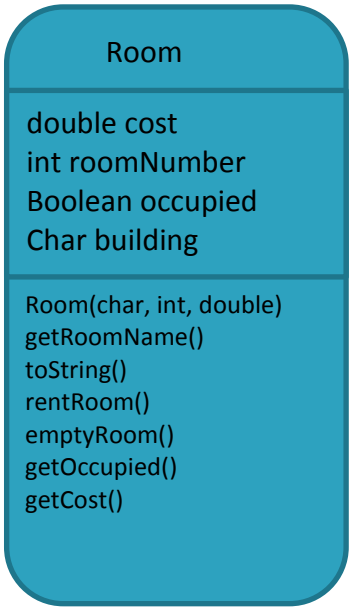
We will want to keep track of the data using the `ResortManagement.java` file as the primary client and building the companion files `Building.java` and `Room.java`.

The main program will then run a loop to allow us to check in (rent a room), check out (be done with a room), and getting data from the files.

## DELIVERABLES

For this assignment, you will be creating two classes that work with `ResortManagement.java` file to implement the necessary behaviour. You are not to adjust/alter `ResortManagement.java` in any way. You will need to implement the two classes `Building.java` and `Room.Java`. Their UMLs are below and a text description follows.




## BUILDING CLASS

This is the object that will maintain a particular building. It will have a particular letter designation ('A', 'B', 'C', …), a size to keep track of how many rooms it has. In addition is will maintain an array of Rooms based upon the size.

It will need to implement the following methods.

### BUILDING(CHAR, INT, DOUBLE)

The constructor will set the size and building letter, and take in the price per room. It will then create the array of rooms of the appropriate size, fill the array with new versions of the room and set the room variables correctly.

Note that while the array starts at zero, the first room should be given the number 1, and so on. So if you have 10 rooms in building Z, you should have Z1, Z2, …. Z9, Z10 as room names.

### TOSTRING()

The `toString()` method for the building will return a string of the following format.

**Building <letter> has <size> rooms : (<numberemptyroom>) empty**.

For example:

```
Building A has 5 rooms :  (4) empty.
```

## PRINTCURRENTSTATUS()

This void method will print out the current status of the building and all of its interior rooms. It should begin by printing the building letter, then on successive lines it should print out each room, the room name, and whether it is currently occupied.

For example:

```
Building B
   Room B1 : is occupied.
   Room B2 : is empty.
   Room B3 : is empty.
   Room B4 : is empty.
```

Note that the rooms are spaced over some amount to show which building they are part of.

## GETVALUE()

This method should return the current double price of all the units that are occupied. So if no rooms are occupied, then the value should be zero.

## LISTOFEMPTY()

This method should return a sting of all the currently empty rooms for this building. So if all the rooms are empty return all the names, and if they are all filled return an empty string.

Example: `A1 A2 A3 A5 A6`

## LISTOFNONEMPTY()

This method should return a sting of all the currently nonempty rooms for this building. So if all the rooms are empty return an empty string, and if they are all filled return all the names. This is the opposite of the method above.

## CURRENTEMPTY()

This method should return an integer number of the number of rooms currently empty.

## RENTROOM()

This is the method used to check out a room. This method should find the first room in the building that is empty, and set it to occupied. Then it should return the name of the room as a String that is to be given out. For example if building A has rooms 1, 2 6, 7 checked out, then it should mark room A3 as now filled and return "A3".

If all the rooms are filled it should return the string "error" in all lower case.

### PUBLIC BOOLEAN CHECKOUT(STRING X)

This method is used to say that people have left a particular room. The input value is the name of the room emptied as a string. The program should find the appropriate room and set it to be un-occupied. If the room is already un-occupied then just reset it to empty. In either case if the room is found return true.

If the room isn't found, return false.

## ROOM CLASS

The Room class is how we manage a single room. Each room is an independent value. Each room should have fields for the building letter it is part of, the cost of the room, the number of the room, and a Boolean value to show if it is currently occupied or not.

### PUBLIC ROOM(CHAR, INT, DOUBLE)

This constructor should set the room letter, the room number and the cost of the room into their respective fields, and make sure the room is initially not occupied.

### PUBLIC STRING GETROOMNAME()

This method should make a string from the building letter and the room number and return it.

### PUBLIC STRING TOSTRING()

This method should return a string that show the room number and the current status of the room in the format **Room <roomname> : is <status>**. The line should look like the following:

```
Room B4 : is empty.
```
or
```
Room B1 : is occupied.
```

### PUBLIC VOID RENTROOM()

This method should set the room to be occupied.

### PUBLIC VOID EMPTYROOM()

This method should set the room to be empty.

### PUBLIC BOOLEAN GETOCCUPIED()
### PUBLIC DOUBLE GETCOST()

These methods should return the appropriate values.

## SAMPLE MAIN PROGRAM EXECUTION

```
What would you like to do?
     1. Rent a room?
     2. Check out a room?
     3. Print a summary
     4. Print a large overview
     0. Quit
1
What building would you like to rent from?
a
Room A1 was rented out.
What would you like to do?
     1. Rent a room?
     2. Check out a room?
     3. Print a summary
     4. Print a large overview
     0. Quit
1
What building would you like to rent from?
a
Room A2 was rented out.
What would you like to do?
     1. Rent a room?
     2. Check out a room?
     3. Print a summary
     4. Print a large overview
     0. Quit
1
What building would you like to rent from?
b
Room B1 was rented out.
What would you like to do?
     1. Rent a room?
     2. Check out a room?
     3. Print a summary
     4. Print a large overview
     0. Quit
2
The currently occupied rooms are : A1 A2 B1
What Room would you like to check out?
a2
Room A2 was cleared.
What would you like to do?
     1. Rent a room?
     2. Check out a room?
     3. Print a summary
     4. Print a large overview
     0. Quit
3
**********************************
** Quick Status of the Resort
** Building A has 5 rooms :  (4) empty.
** Building B has 4 rooms :  (3) empty.
** Building C has 6 rooms :  (6) empty.
** Building D has 10 rooms :  (10) empty.
```

```
**********************************
** The currently occupied rooms are : A1 B1
** The current value of the resort is $400.00.
** There are   23 empty rooms.
**********************************
What would you like to do?
     1. Rent a room?
     2. Check out a room?
     3. Print a summary
     4. Print a large overview
     0. Quit
4
**********Expanded Status of the Resort*******
Building A
   Room A1 : is occupied.
   Room A2 : is empty.
   Room A3 : is empty.
   Room A4 : is empty.
   Room A5 : is empty.
Building B
   Room B1 : is occupied.
   Room B2 : is empty.
   Room B3 : is empty.
   Room B4 : is empty.
Building C
   Room C1 : is empty.
   Room C2 : is empty.
   Room C3 : is empty.
   Room C4 : is empty.
   Room C5 : is empty.
   Room C6 : is empty.
Building D
   Room D1 : is empty.
   Room D2 : is empty.
   Room D3 : is empty.
   Room D4 : is empty.
   Room D5 : is empty.
   Room D6 : is empty.
   Room D7 : is empty.
   Room D8 : is empty.
   Room D9 : is empty.
   Room D10 : is empty.
#####
#####
What would you like to do?
     1. Rent a room?
     2. Check out a room?
     3. Print a summary
     4. Print a large overview
     0. Quit
0
```

## PROGRAM NOTES

While you are not supposed to alter `ResortManagement` in any way, I can't stop you from commenting out parts of the code as you test it. As long as the file works with your files at the end, then you are ok. So make the programs work in stages as you go. Get individual methods working first.

Use plenty of println statements as you go to see what is working and what isn't working. Printing out test variables can be very useful as you go.