

# CISC121 Winter 2023 - Assignment 5

Due: Wednesday March 29th 11:59pm (EDT)

Notes:

- Your assignment should follow the python style guide, and any other style/commenting expectations as discussed in lecture
- By submitting the assignment, you agree that you have followed the Queen's Academic Integrity policy, and that your assignment was completed independently.

## Scenario

You are working on creating a data storage model for a local hotel. Each day they have people arriving who will be booked into different types of rooms: suites, standard, and executive suites they have branded as Taylor Suites.

The hotel rooms each have a room number stored as an integer and are marked as either available or unavailable (true or false). Each room is marked as unavailable until the check out date for the guests. Each different suite type has an available date. These rooms are difficult to clean, so once a guest checks out, the room will not be available until 1 day later for the suites, and 2 days later for the Taylor Suites, standard rooms are available on the same day. [Hint: The availability date is different for each room type].

Every person arrives with a check out date. They are assigned to a type of room. The hotel keeps track of which guests are in each room, and when the check out date for each room is. There are 19 suites, 89 standard rooms, and 13 Taylor Suites. Each guest has a name, a check out date, a list of services they receive (such as room service, or movies), and a total bill (an integer value) to be given at the end of their stay.

The services are also stored in this system as a class and contain a service name and a cost for that particular service.

We can summarize this information in the following class structure:

- Hotel Room
  - Suite (Room Subclass)
  - Standard room (Room Subclass)
  - Taylor Suite (Room Subclass)
- Guest
- Service

## Question 1

- a. (8 marks) Define the classes as listed above by writing the init functions for all classes and subclasses.
- b. (12 marks) Create the following methods for the given classes. The docstrings for these methods are in the appendix. You must follow the parameters and returns for these functions as indicated.
  - Hotel Room
    - Guest Check-in: Set the room to be unavailable from the start day of their stay until the room can be cleaned following the guest checkout
  - Guest
    - Add service
    - Send bill

## Question 2 – 5 marks

Consider the documentation for Stacks that was provided in class and through the worksheet. Add the following method to the class STACK.

```
def reverse(self):
    """
    -----
    Reverses the contents of the source stack
    Use: source.reverse()
    -----
    Returns:
        None
    -----
    """
```

## To Submit:

Submit the following files as a **SINGLE zip file** named **a5\_999999999.zip (your student number)**:

1. **hotel\_classes.py**: A file containing all of your classes for both questions.
2. **Stack\_class.py**: your stack class file
3. **a5\_q1.py**: the main program for the first question
4. **a5\_q2.py**: the main program for the second question
5. **testing.txt**: Include all tests you run, and the output. Simply state what the resulting set of input was, and then the corresponding output. Whenever your tests fail, you may include notes to the TAs. It is encouraged to discuss *why* you've chosen each test case, but not required. It is good to have 3-5 tests for each function. You should be testing your code much more often than this.

## Appendix – Docstrings

```
def guest_check_in(self, some_guest):
    """
    -----
    Checks a guest into a room and updates availability
    -----
    Parameters:
        Some_guest - the guest checking in (guest object)
    Returns:
        None
    -----
1
2
3
4
5
6
7
8
9
10
11
12

def add_service(self, service_name, service_cost):
    """
    -----
    Adds a service to the bill of a guest
    -----
    Parameters:
        Service name - name of service (str)
        Service_cost - cost of the service (int)
    Returns:    None
    -----
1
2
3
4
5
6
7
8
9
10
11
12

def send_bill(self):
    """
    -----
    Based on the total services and bill total provides a
        Printed statement of all services with the total cost
    -----
    Parameters:    None
    Returns:       None
    -----
1
2
3
4
5
6
7
8
9
```