This document has been created to help you organize the screenshots you need to capture to demonstrate the completion of the **mini03_reservations** project. When you are done with all screenshots, save a PDF version of the file and upload it to Canvas.

The overloaded = operator for the Date class

In the following space, capture screenshot(s) that demonstrate the ***= operator*** has been overloaded to properly set the data members for **at least 5 different dates**. These dates must cover different months of a year and include some dates such as the first, middle, and last day of a month. Use the **overloaded <<** operator to help you generate sample output.

```
The date 1/1 is 0 day(s) from January 1

The date 3/15 is 73 day(s) from January 1

The date 6/30 is 180 day(s) from January 1

The date 9/1 is 243 day(s) from January 1

The date 12/31 is 364 day(s) from January 1
```

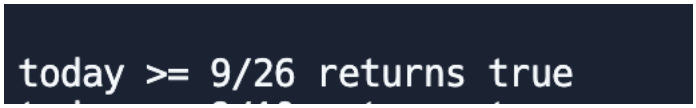
The day_of_the_week member function of the Date class

In the following space, capture screenshot(s) that demonstrate this function works properly. Use the ***= operator*** to generate **at least 5 different dates** and display their corresponding days.

The overloaded >= operator for the Date class

In the following space, capture screenshot(s) that demonstrate the **>= *operator*** has been properly overloaded for the following cases. Use proper output statements to clearly explain the dates being compared and the results of comparisons. It's important to test the following scenarios:

- The Date object and the string represent the same month and day.



- The Date object and the string represent two different days in the same month and the result of the comparison is true.

- The Date object and the string represent two different days in the same month and the result of the comparison is false.

- The Date object and the string represent the same day, but different months and the result of the comparison is false.

- The Date object and the string represent the same day, but different months and the result of the comparison is true.

- The Date object and the string represent different day and month and the result of the comparison is true.

```
today >= 8/10 returns true
```

- The Date object and the string represent different day and month and the result of the comparison is false.

```
today >= 10/10 returns false
```

The overloaded <= operator for the Date class

In the following space, capture screenshot(s) that demonstrate the **<= *operator*** has been properly overloaded for the following cases. Use proper output statements to clearly explain the dates being compared and the results of comparisons. It's important to test the following scenarios:

- The Date object and the string represent the same month and day.

```
today <= 9/26 returns true
```

- The Date object and the string represent two different days in the same month and the result of the comparison is true.

```
today <= 9/30 returns true
```

- The Date object and the string represent two different days in the same month and the result of the comparison is false.

```
today <= 9/24 returns false
```

- The Date object and the string represent the same day, but different months and the result of the comparison is false.

```
today <= 8/26 returns false
```

- The Date object and the string represent the same day, but different months and the result of the comparison is true.

```
today <= 10/26 returns true
```

- The Date object and the string represent different day and month and the result of the comparison is true.

```
today <= 10/10 returns true
```

- The Date object and the string represent different day and month and the result of the comparison is false.

```
today <= 8/10 returns false
```

The constructor for the Reservation class

In the following space, capture screenshot(s) that demonstrate the ***constructor*** is working properly by creating **two different Reservation objects** and use the ***to_string*** function to help generate sample output.

```
Reservation Details:
Reservation Valentines Day Reservation on The dat
e 2/14 is 44 day(s) from January 1

Reservation Details:
Reservation Christmas Day Reservation on The date
 12/25 is 358 day(s) from January 1
```

The is_high_season member function of the Reservation class

In the following space, capture screenshot(s) that demonstrate the i**s_high_season** is working properly by creating **at least 3 different Reservation objects** and display whether they fall in the high season or not. Make sure to choose some dates before Dec 20, after Dec 20, and before Jan 4.

```
res1 12/2 is not a high season reservation

res2 12/25 is a high season reservation

res3 1/1 is a high season reservation
```

The constructor for the HotelReservation class

In the following space, capture screenshot(s) that demonstrate the ***constructor*** is working properly by creating **two different HotelReservation objects** and use the ***to_string*** function to help generate sample output.

```
Hotel Reservation Details:
Hotel Reservation on The date 2/14 is 44 day(s)
 from January 1 at Conrad Hilton for 1 night(s)


Hotel Reservation Details:
Hotel Booking on The date 12/25 is 358 day(s) f
rom January 1 at Waldorf Astoria for 3 night(s)
```

The calculate_cost member function of the HotelReservation class

In the following space, capture screenshot(s) that demonstrate the calculate_cost function is working properly for reservations whose dates fall in the high season or regular season and with a different number of nights.

```
Hotel Reservation Details:
Hotel Reservation on The date 2/14 is 44 day(s)
 from January 1 at Conrad Hilton for 1 night(s)
Total cost: $999


Hotel Reservation Details:
Hotel Booking on The date 12/25 is 358 day(s) f
rom January 1 at Waldorf Astoria for 3 night(s)
Total cost: $5850
```