# CST8132 Lab 6: Dinner & A Movie

## Program Description

You are planning a romantic evening for two when your date invites along another couple. When the night of the big outing rolls around, not everyone is able to make it due to the unpredictable weather. As the organizer you will definitely be in attendance, but you don’t know until you’re already out that you might be the only one who makes it out of the house that night. There will randomly be between 1 – 4 **guests** on the date night. You can assume that if only two people are going, it is you and your date.

Your evening plans consist of dinner and a movie, however you cannot decide what movie to go to or what time. As a group, you narrow the options down to 3 movies, all playing at either 6:00pm or 10:00pm. You **randomly** pull a movie title out of a hat and flip a coin to decide which of the two movie times to attend.

The **restaurant** servers a variety of **menu items**. The menu is broken up into 4 menu item types – **Appetizers**, **Entrees**, **Drinks**, and **Desserts**. You all agree to order 1 appetizer for the table, you each order your own entrée and drink, and each couple (or person flying solo) orders a dessert to share. In total, there should be 1 appetizer, 1 – 4 entrees, 1 – 4 drinks, and 1 – 2 desserts ordered by the end of the meal. Each **order** belongs to a specific **guest**. You’ve ordered the appetizer, as the host, so it should be part of your order. The desserts should be added to odd-numbered orders (assuming there are between 1 – 4 orders). At the end of the dinner, you will each be presented with your own detailed bill to pay.

The restaurant also has **happy hour** between 5 – 7pm. Note that if you go to the 6:00pm movie, you will have to eat after the movie, so you’ll miss out on the happy hour specials. Otherwise, if you are going to the late show, you need to apply happy hour discounts to your bills. Discounts should be displayed on the final bill, along with the original prices.



# Implementation

1. Create a new **Eclipse** project.
2. Add a folder named **lib** to your project.
3. Import the file **CST8132\_Restaurant.JAR** to the **lib** folder of your project.
4. Add the JAR file to your Build Path.
5. Create a new **Package** named **dateNight**.

# Package: dateNight

## Class: Entree extends MenuItem

1. Imported Classes
   1. cst8132.restaurant.MenuItem
2. Methods
   1. **public Entree(String name, double price)**
      1. Call the **super** constructor, passing the **name** and **price** parameters.

## Class: Dessert extends MenuItem

1. Imported Classes
   1. cst8132.restaurant.MenuItem
2. Methods
   1. **public Dessert(String name, double price)**
      1. Call the **super** constructor, passing the **name** and **price** parameters.

## Class: Bill

1. Imported Classes
   1. java.util.ArrayList
   2. java.util.HashMap
   3. cst8132.restaurant.Appetizer
   4. cst8132.restaurant.Drink
   5. cst8132.restaurant.MenuItem
2. Instance Variables
   1. boolean isHappyHour
   2. HashMap<String,ArrayList<MenuItem>> orders
   3. double subtotal
   4. double hstRate
   5. final int maxMenuItemLength
3. Methods to Implement
   1. **public boolean getHappyHour()**
      1. Return the value of **isHappyHour**.
   2. **public void setHappyHour(Boolean isHappyHour)**
      1. Set the value of **isHappyHour**.
   3. **public HashMap<String, ArrayList<MenuItem>> getOrders()**
      1. Return the value of the **orders** variable.
   4. **public double getSubtotal()**
      1. Return the value of **subtotal**.
   5. **public double getHst()**
      1. Return the calculated HST value.
      2. HST = (subtotal – happyHourDiscount) \* hstRate
   6. **public double getHstRate()**
      1. Return the value of the **hstRate**.
   7. **public double getTotal()**
      1. Return the calculated total.
      2. Total = subtotal – happyHourDiscount + hst
   8. **public double getHappyHourDiscount()**
      1. Return the calculated happy hour discount, or 0 if no discount should be applied.
      2. Using the **instanceof** type comparison operator, apply the appropriate discounts for each Appetizer or Drink ordered. Discounts are described in requirements above.
      3. **Hint –** you can use nested enhanced for loops to iterate over the HashMap and ArrayList.
4. Implemented Methods
   1. The following methods have already been implemented – they do not need to be modified, however you must add **Javadoc** **and comments** to them to indicate your understanding of the methods.
      1. public String toString()
      2. public boolean addOrderItem(String guest, MenuItem item)

## Class: DoubleDate

1. Imported Classes
   1. java.util.ArrayList
   2. java.util.Arrays
   3. java.util.Random (optional)
   4. cst8132.restaurant.Menu
   5. cst8132.restaurant.Restaurant
2. Instance Variables
   1. ArrayList<String> guests
   2. String[] movies
   3. String movieTitle
   4. int movieTime
   5. Restaurant restaurant
   6. Menu menu
   7. Bill bill
3. Class Variables
   1. Random random
4. Methods
   1. **public DoubleDate(String yourName, String… guests)**
      1. Initialize the **guests** instance variable by creating a **new ArrayList<String>**. Give your ArrayList a default capacity of 4.
      2. Using the **ArrayList** **add** method, add your name to the **ArrayList**. This will insert your name in position 0 of the **ArrayList**.
      3. If there are any additional guests to add, use the **ArrayList addAll** and **Arrays.<String>asList**(array) methods to add the optional guests to the **ArrayList**.
      4. Initialize the **restaurant** by calling the static **Restaurant.getInstance(String name)** method.
      5. Initialize the **menu** by calling the **restaurant.getMenu()** method, and then invoking the **addMenuItems()** method.
      6. Initialize the **bill** by calling the default constructor of the **Bill** class.
      7. Initialize the **movies** array by adding at least 3 movie titles to the array.
   2. **public String pickAMovie(String[] movies)**
      1. Using either the **Math.random** or **Random.nextInt** method, select a movie to attend, and return this value.
   3. **public int getShowing()**
      1. Using either the **Math.random** or **Random.nextInt** method, randomly return the value 6 or 10, to represent the time you will be attending the movie.
   4. **public void addMenuItems()**
      1. There is a method in the Menu class with the following signature:

**public boolean addMenuItem(String itemType, String name, double price)**

* + 1. Using this method and the **itemType** values listed below, add at least 3 items of each type to your menu. **Drink** prices should be at least $5.00. **Dessert** and **Appetizer** prices must be evenly divisible by 2.
       1. Drinks
       2. Desserts
       3. Appetizers
       4. Entrees
    2. Menu item names should be a maximum of 30 characters in length for optimal formatting of outpout.
  1. **public boolean placeOrder(String guest, String itemType)**
     1. Get a random **MenuItem** to order by calling the **getRandomMenuItem(String itemType)** method of the **Menu** class.
     2. Add this **MenuItem** to this person’s order by calling the **addOrderItem(String guest, MenuItem menuItem)** method of the **Bill** class.
     3. The **addOrderItem** method returns a **boolean** value, which should be returned by this **placeOrder** method.
  2. **public static void main(String[] args)**
     1. Initialize a new **DoubleDate** by passing name(s) to the constructor. The first name passed should be your own. The second name would be your date. The third name would be your date’s friend and the fourth name passed would be their date. Note that only the first name is mandatory.
     2. Pass a **String array** of at least 3 movie titles (enter your own set of titles) to the **pickAMovie** method and assign the return value to the **movieTitle** instance variable.
     3. Call the **getShowing** method and assign the return value to the **movieTime** variable.
     4. Based on the **movieTime**, set the **isHappyHour** variable of your **Bill** by calling the **setHappyHour(boolean isHappyHour)** method.
     5. For each **guest** at the restaurant, call the **placeOrder(String guest, String itemType)** method as needed, based on the requirements in the program description. This will add random menu items to your bill.
     6. Print the **DoubleDate** object by implicitly calling its **toString()** method, like this:

**System.out.println(date);**

* 1. **public String toString()**
     1. Output the list of movie options, the selected movie, and show time.
     2. Output the name of the restaurant, and based on the show time, whether you will be meeting there before or after the movie.
     3. Output **one of** the following statements:

It's happy hour! $2 off drinks, and 1/2 price appetizers!!

or

We'll be missing happy hour, but we'll still be happy!

* + 1. Output the menu, by calling its **toString()** method.
    2. Output the bill, by calling its **toString()** method.

# Package: cst8132.restaurant

## Class: Restaurant

## Class: Menu

## Class: MenuItem

## Class: Drink extends MenuItem

## Class: Appetizer extends MenuItem

# Marking Scheme

TBA

# Junit

TBA

# Swing

TBA

# Submission Requirements

TBA