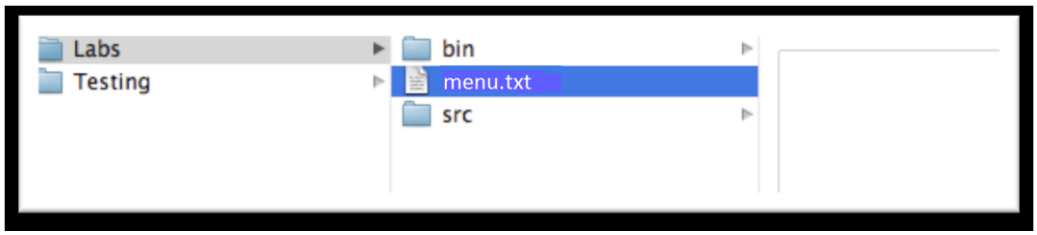
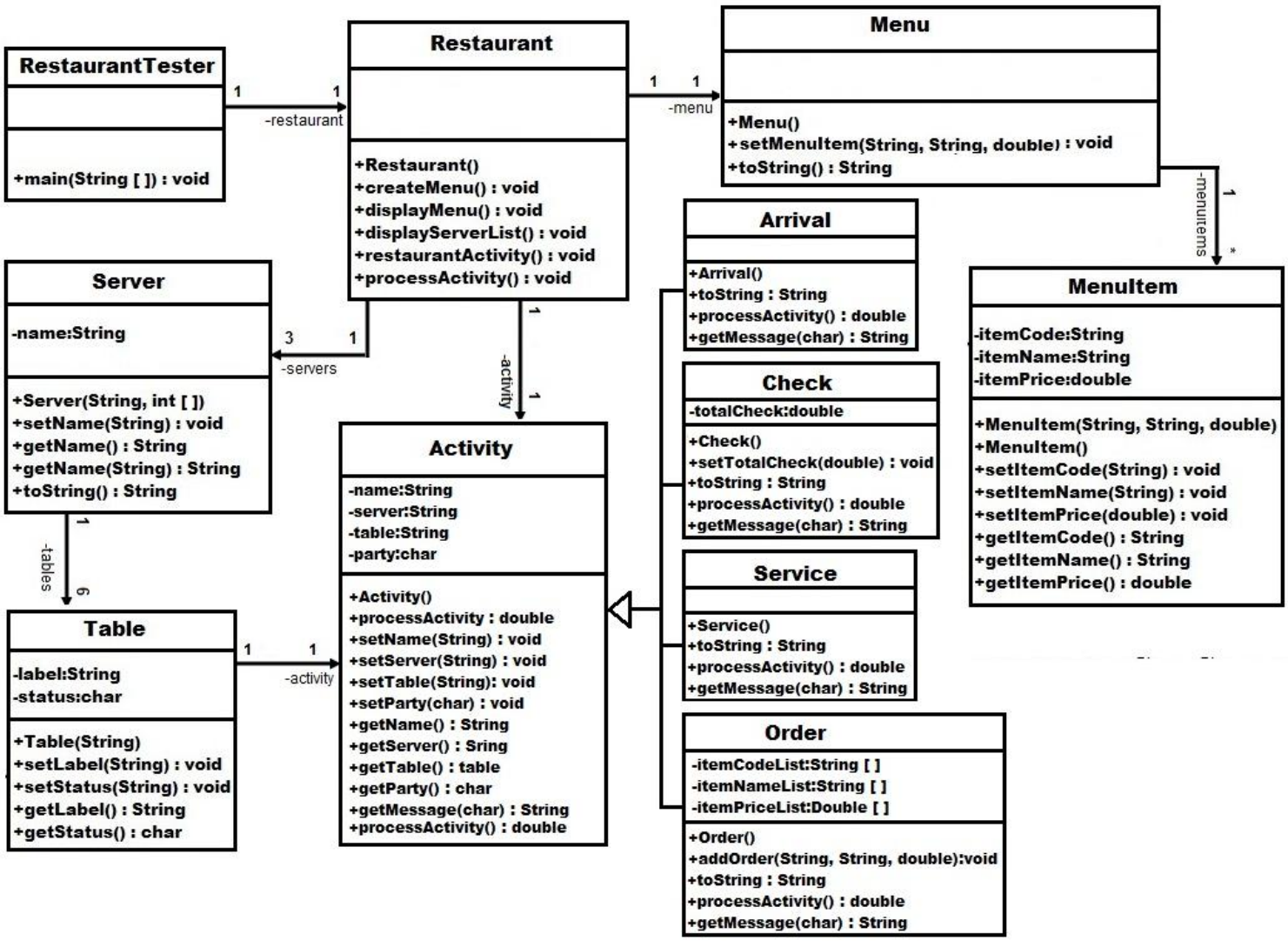
Semester Project Part 1
Restaurant Activity with Inheritance

The objective of this assignment is to develop an understanding into the most powerful concepts of Object Oriented Programming called, inheritance and polymorphism.

For this lab, you will need to place the menu.txt file included with the instructions in the Eclipse workspace directory just above the bin and src folders where your program is contained. e.g.,





For inheritance and polymorphism, classes Arrival, Check, Service, Order are created as the subclasses of Activity class. The methods toString, processActivity() and getMessage() are overridden in the child classes and need to simulate polymorphism.

```
Waiters: first name followed by table list
John 1,2,5,9,11,15
Maria 3,4,6,7,17,18
Mike 8,10,12,13,14,26
```

```
Menu: listing of the full menu: item code, name, price
A1 Bruschetta 5.29
A2 Caprese_Flatbread 6.10
A3 Artichoke-Spinach_Dip 3.99
A4 Lasagna_Fritta 4.99
A5 Mozzarella_Fonduta 5.99
E1 Lasagna_Classico 6.99
E2 Capellini_Pomodoro 7.99
E3 Eggplant_Parmigiana 8.99
E4 Fettuccine_Alfredo 7.49
E5 Tour_of_Italy 14.99
D1 Tiramisu 2.99
D2 Zeppoli 2.49
D3 Dolcini 3.49
S1 Soda 1.99
S2 Bella_Limonata 0.99
S3 Berry_Acqua_Fresca 2.88
```

NOTES:
Each program should include comments that explain what each block of code is doing. Additionally, the programs should compile without errors, and run with the results described in the exercise. The following deductions will be made from each exercise if any of the following is incorrect or missing:

- Proper formatting [5 points]
- Proper names for classes and variables [5 points]
- Comments [5 point]
- Program doesn't compile [10 points]
- Source code (java file) missing [10 points]
- Executable (class file) missing [10 points]
- Missing or wrong instance variable declarations [5 points each]
- Missing or wrong methods [5 points each]

TURN IN TO BLACKBOARD (i.e. submit):
A single zip file (**Project1.zip**) containing the following:

- Source code files: RestaurantTester.java, Restaurant.java, Menu.java, MenuItem.java, Table.java, Server.java, Activity.java, Arrival.java, Service.java, Check.java, Order.java
- Class files: RestaurantTester.class, Restaurant.class, Menu.class, MenuItem.class, Table.class, Server.class, Activity. class, Arrival. class, Service. class, Check. class, Order. Class

This lab is due on or before 6:00am Saturday October 26, 2013

Class: RestaurantTester	
Method	**Description**
main	**Creates the Restaurant object; presents options to users to make choices. The main method needs to be coded to support the following options menu – and based on the choice; it needs to call specific methods in the Restaurant class:** Welcome to Java Restaurant 1. Display menu 2. Display Server List 3. Restaurant Activity 4. Quit When user enters choice as 4 then it brings up the sub menu of the Restaurant Activity as below: Enter choice: 4 1. Restaurant activity 2. Quit

Class: Restaurant	
Method	**Description**
Restaurant	**Constructor. Creates the object of Menu and and ArrayList of Server objects. It reads the file and send the sends the code, name and price to setMenuItem() method of the Menu class.**
createMenuItems	**Creates the objects of MenuItems and adds those to the Menu**
displayMenu	**Displays the Menu by listing all the MenuItems which were created in createMenuItems method**
displayServerList	**Displays the list of Servers in the Restaurant**
restaurantActivity	**Interactively reads the activities and calls the processActivity method to display output based on the activity. This lab is adding to the functionalities of the previous lab. The typical set of rules are as follows:**

	T1 P3 → a party of 3 arrives and assigned to table 1. **T29 P4 → a party of 4 arrives, but cannot be assigned to table T29 since T29 does not have a server associated with it.** **T1 O A1 A2 R3 → Table T1 is placing the order; since R3 is not a valid, code the user needs to be informed R3 is not part of the menu.** **Take a look at the example output which tried few possibilities of a restaurant activity.**
processActivity	**Use the split() method in the String class to tokenize the activity. For example, if "T1 A2" is the activity then the split method based on the activity will create an array of 2 elements, T1 and A2.** **Here is the example statement:** String [] element = activity.split("\\s"); will split based on space between the codes. If activity is "T1 P2" then we will produce the following array: element[0] = T1 element[1] = P2 List of service codes are: P – Customers to be seated O – Order has arrived S – Food is serverd C – Table is checking out T1 P1 – Table 1 is seating Party of 1 T2 O A1 A2 E1 – Table 2 is placing the order for menuitems A1, A2, and E1 T3 S – Food is being served at Table T3 T1 C – Table T1 is checking out Note: In this lab, you will need to maintain the status of the Table. For example: food must be ordered before it is served; the order needs to be placed first. Similalry, a table cannot checkout if they have not ordered food or the food had not been served it. A typical sequence for a given Table T is P, O, S and then C which will put the status of table from initially being available into O(ccupied), OR(dered), S(erved) and back to A(vaialble). T1 P1 → moves the status to 'O' for Occupied T1 S → not possible because T1 has not placed order T1 C → not possible because the Food has been served T1 O A1 → this will change the status to 'R' for Ordered; and an Order Object will be created to keep track of the Order because at the time of the activity as T1 C → the total check has to be printed. A typical Order will include the name of the menu item and the price.

Class: Menu	
Method	**Description**
Menu	Constructor. Creates the Menu object and instantiate the ArrayList of MenuItems
setMenuItem	Setter method to add MenuItem
getValue	Getter method for the value attribute
getName	Getter method for the name attribute
toString	Creates a list of MenuItems as a String

Class: MenuItem	
Method	**Description**
MenuItem	Constructor. Creates a MenuItem object
setItemCode	Setter method for the itemCode attribute
setItemName	Setter method for the itemName attribute
setItemPrice	Setter method for the itemPrice attribute
getItemCode	Getter method for the itemCode attribute
getItemName	Getter method for the itemName attribute
getItemPrice	Getter method for the itemPrice attribute

Class: Table	
Method	**Description**
Table	Constructor. Creates a Table object; and an Activity object to keep track of the activities per Table per Server
setLabel	Setter method for the label attribute
setStatus	Setter method for the status attribute
getLabel	Getter method for the label attribute
getStatus	Getter method for the status attribute

Class: Server	
Method	**Description**
Server	Constructor. Creates a Server object and instantiates the ArrayList of Table objects
setName	Setter method for the name attribute
getName	Getter method for the name attribute
getStatus	Searches the Server name for a give table

Class: Activity	
Method	**Description**
Activity	Constructor. Creates a Activity object and keeps track of the activity name, the server name, the specific table and the number of people in the part T1 P3 → Arrival, John, T1 and 3 for the Arrival activity, Table T1 is assigned to John. T1 O A1 A2 → Order T1 S → Service T1 C → Check
setName	Setter method for the name attribute
setServer	Setter method for the server attribute

setTable	Setter method for the table attribute
setParty	Setter method for the party attribute
getName	Getter method for the name attribute
getServer	Getter method for the server attribute
getTable	Getter method for the table attribute
getParty	Getter method for the party attribute

Class: Arrival	
Method	**Description**
Arrival	Constructor. Creates an Arrival object; calls the setName() method of the parent class
toString	Displays the detail of the activity; for example, the Party of 3 has arrived and has been assigned to table T1 for the activity "T1 P3"
processActivity	Any processing related to Arrival
getMessage	Message related to Arrival object

Class: Order	
Method	**Description**
Order	Constructor. Creates an Order object; calls the setName() method of the parent class; initializes the lists of Order Code, Name, and Price
toString	Displays the detail of the activity; for example, table T1 has placed an order for the activity "T1 O A2 A3"
processActivity	Any processing related to Order
getMessage	Message related to Order object
addOrder	Adds code, name and price for new orders

Class: Check	
Method	**Description**
Check	Constructor. Creates a Check object; calls the setName() method of the parent class;
toString	Displays the detail of the activity; for example, table T1 is checking out for the activity "T1 C"
processActivity	Any processing related to Check
getMessage	Message related to Check object
setTotalCheck	Receives the total cost of the order to be used in displaying the check at the time check out

Class: Service	
Method	**Description**
Server	Constructor. Creates a Server object and instantiates the ArrayList of Table objects
toString	Displays the detail of the activity; for example, the food is served at table T1 for the activity "T1 S"
processActivity	Any processing related to Sercice
getMessage	Message related to Service object

SAMPLE OUTPUT

Welcome to Java Restaurant

****
1. Display Menu
2. Display Servers List
3. Restaurant Activities
4. Quit
****
Enter choice: 1

Code	Name	Price
A1	Bruschetta	5.29
A2	Caprese_Flatbread	6.1
A3	Artichoke-Spinach_Dip	3.99
A4	Lasagna_Fritta	4.99
A5	Mozzarella_Fonduta	5.99
E1	Lasagna_Classico	6.99
E2	Capellini_Pomodoro	7.99
E3	Eggplant_Parmigiana	8.99
E4	Fettuccine_Alfredo	7.49
E5	Tour_of_Italy	14.99
D1	Tiramisu	2.99
D2	Zeppoli	2.49
D3	Dolcini	3.49
S1	Soda	1.99
S2	Bella_Limonata	0.99
S3	Berry_Acqua_Fresca	2.88
C1	Coffee	1.0
C2	Soda	1.0
C3	Milk	1.0

****
1. Display Menu
2. Display Servers List
3. Restaurant Activities
4. Quit
****
Enter choice: 2

Name	Tables
John	T1(A) T2(A) T5(A) T9(A) T11(A) T15(A)
Maria	T3(A) T4(A) T6(A) T7(A) T17(A) T18(A)
Mike	T8(A) T10(A) T12(A) T13(A) T14(A) T16(A)

****
1. Display Menu
2. Display Servers List
3. Restaurant Activities
4. Quit
****
Enter choice: 3

****
1. Restaurant activity
2. Quit
****
Enter choice: 1
Enter activity: T1 P4

Staus of table T1: Available
Party of 4 has been assigned to Table T1
Welcome! your server John will be there soon to take care of you.
Table T1 is now occupied

****
1. Restaurant activity
2. Quit
****
Enter choice: 1
Enter activity: T5 P3

Staus of table T5: Available
Party of 3 has been assigned to Table T5
Welcome! your server John will be there soon to take care of you.
Table T5 is now occupied

****
1. Restaurant activity
2. Quit
****
Enter choice: 1
Enter activity: T9 P4

Staus of table T9: Available
Party of 4 has been assigned to Table T9
Welcome! your server John will be there soon to take care of you.
Table T9 is now occupied

****
1. Restaurant activity
2. Quit
****

Enter choice: 1
Enter activity: T1 S
Table T1:
Status of table is Occupied!
Food has not been ordered

****
1. Restaurant activity
2. Quit
****
Enter choice: 1
Enter activity: T1 C
Testing: O
Table T1:
Status of table is Occupied!
Food has not been ordered. No checkout required**

****
1. Restaurant activity
2. Quit
****
Enter choice: 1
Enter activity: T1 O A2 A3 R4**

**Order: table T1 has placed an order!
Your server John will be notified**

**Caprese_Flatbread 6.1
Artichoke-Spinach_Dip 3.99
Item Code: R4 is not part of the menu
****
1. Restaurant activity
2. Quit
****
Enter choice: 1
Enter activity: T1 C
Testing: R
Table T1:
Status of table is Ordered!
Food has not been served. No checkout required**

****
1. Restaurant activity
2. Quit
****
Enter choice: 1
Enter activity: T1 S

Service:
Food arrives at table: T1

****
1. Restaurant activity
2. Quit
****
Enter choice: 2

****
1. Display Menu
2. Display Servers List
3. Restaurant Activities
4. Quit
****
Enter choice: 2

Name Tables
John T1(S) T2(A) T5(O) T9(O) T11(A) T15(A)
Maria T3(A) T4(A) T6(A) T7(A) T17(A) T18(A)
Mike T8(A) T10(A) T12(A) T13(A) T14(A) T16(A)
****
1. Display Menu
2. Display Servers List
3. Restaurant Activities
4. Quit
****
Enter choice: 3

****
1. Restaurant activity
2. Quit
****
Enter choice: 1
Enter activity: T1 C
Testing: S

Caprese_Flatbread 6.1
Artichoke-Spinach_Dip 3.99
Item Code: R4 is not part of the menu

Party at table: T1 gets the check, pays and leaves.

Check:
Total: 10.090
Thanks!

T1 is now Available

****
1. Restaurant activity
2. Quit
****
Enter choice: 2

****
1. Display Menu
2. Display Servers List
3. Restaurant Activities
4. Quit
****
Enter choice: 4
Thank you, the Java Restaurant is now closed.