Homework 3

Object Oriented Programming CSI 405

Problem 1 (edu.albany.cruise, edu.albany.cargo)

Model Ship, CruiseShip and CargoShip classes

Design a ship class that has the following members:

1. A field for the name of the ship (string)
2. A field for the year that the ship was built (string)
3. A constructor and appropriate accessors and mutators
4. A toString method that displays the ship's name and the year it was built

Design a CruiseShip class that extends the Ship class. The CruiseShip class should have the following members:

1. A field for the maximum number of passengers (int)
2. A constructor and appropriate accessors and mutators
3. A toString method that overrides the toString method in the base class. The CruiseShip class's toString method should display only the ship's name and the maximum number of passengers

Design a CargoShip class that extends the Ship class. The CargoShip class should habe the following members:

1. A field for the cargo capacity in tonnage (int)
2. A constructor and appropriate accessors and mutators
3. A toString method that overrides the toString method in the base class. The CargoShip class's toString method should display only the ship's name and the ship's cargo capacity

Demonstrate the classes in a program that has a Ship array. Assign various Ship, CruiseShip and CargoShip objects to the array elements. The program should then step through the array, calling each object's toString method.

Problem 2 (edu.albany.*your_package_names)*

Create *packages* as needed per design

Create classes modeling the purchase of sandwich / sandwiches at a typical sandwich shop. Use three objects: Cashier, Customer, SandwichMaker. (You can add more classes if you want but these three should be present in the project)

Follow the steps to complete the homework:

1. Create a sequence / flow diagram – This diagram would show how an interaction is initiated. Also it shows which class will initiate which other class
2. Indicate what information (if any) is passed between the classes
3. Figure out what instance variables will be needed in each of the classes

4. Create UML diagrams - The UML diagrams should follow the exact formatting shown in class slides. Submit UML diagrams in a separate folder as images with your code submission.
5. Create all the classes
6. Create a driver class (SandwichDriver) to test the functionality
7. Handle all possible errors with try catch blocks wherever necessary

Note: There is no one solution for the homework. Each person can think differently.

Grading

Implementation	40%
Execution	40%
UML diagrams	10%
Code clarity / organization	10%
Sequence diagram	5%

Submission

Check exact deadlines on blackboard. Deadlines are followed strictly when grading. No excuses. Check deadline policy in syllabus

Submit any specific instructions needed for grading/executing your homework.
Your project(s) should be a 7-zip (or any other zip format) file submitted on blackboard. It should be an eclipse project that the grader can import into his/her eclipse environment and execute. If this criterion is not followed you will lose points.

The UML diagrams should be submitted separately in the same submission as a folder of images well labelled/organized. (Not inside the project archive). Same for sequence diagram(s).
Follow exact deadlines, timing on blackboard. Rules for late submission apply as per syllabus.

Follow naming conventions posted on blackboard.