**44-542 Object-Oriented Programming**

**Lab07: Inheritance and Polymorphism**

**Objective:** Covers the usage of **Inheritance, Polymorphic Substitution and Late binding Polymorphism.**

**NOTE:**

* Do not hard code any values.
* Check the sample output to know how the results need to be printed.
* Read every instruction carefully and follow them strictly.
* Do not change the name of the attributes, and methods given below.
* @author annotation must contain your full name for all the classes in this project.
* Use @override annotation for every relevant method in all the classes of your project.
* Do not forget to indent your code.
* Use only the concepts covered till last class.
* Generate Javadoc before uploading your project.

**Overview:**

In this lab, you have to create all the classes given in UML diagram. The input text file is provided to you. You have to define the relationships from the below given UML diagram. Generate Javadoc for your project which must include appropriate comments, @param, @return annotations for all constructors and methods.

1. Create a project and name it as **Lastname\_lab07InheritanceAndPolymorphism** where Lastname is your last name.
2. Define the classes as shown below in the UML diagram.
3. The description for each class is provided after the UML diagram.

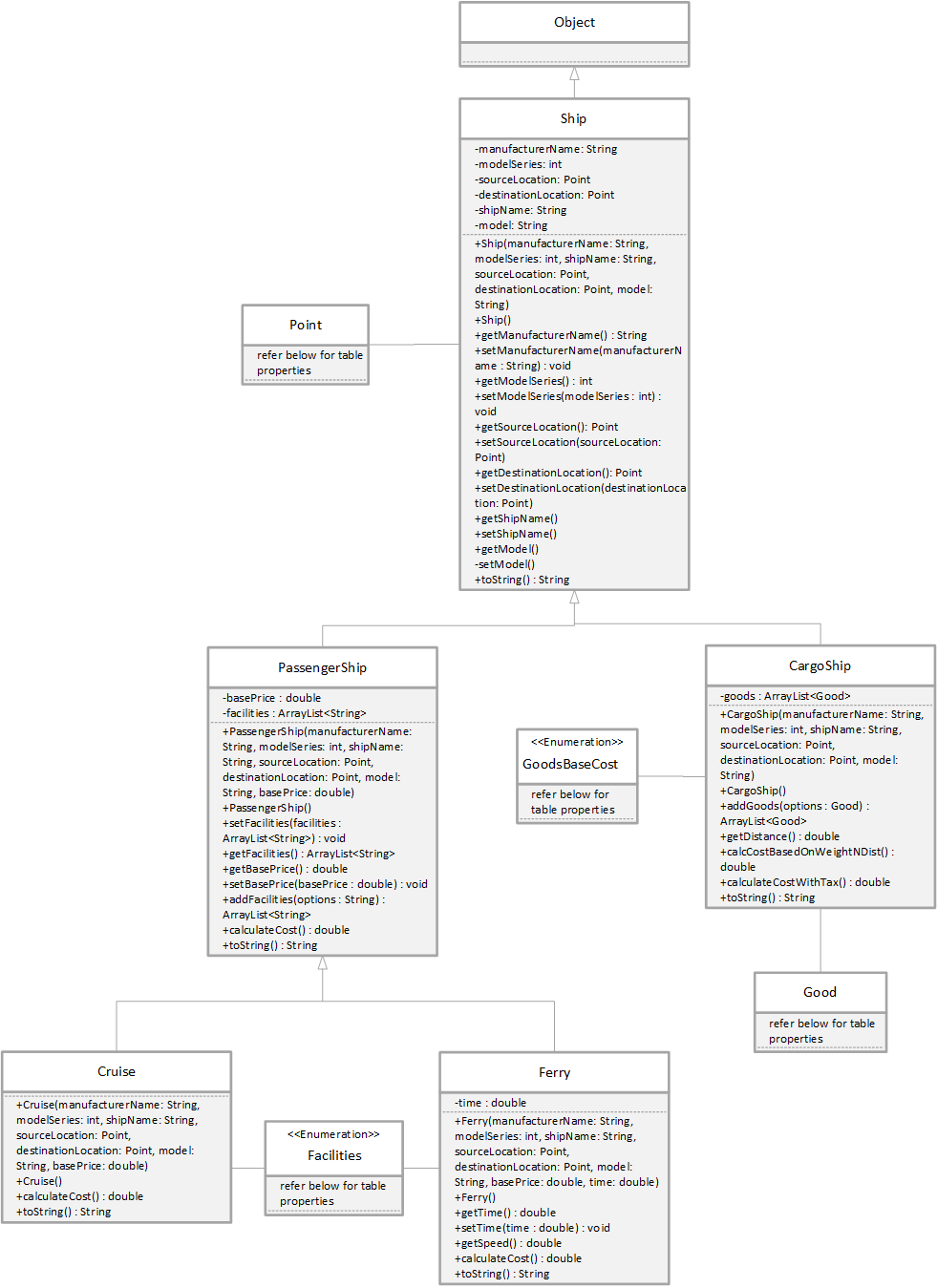
**Description:** A ship is classified into two types, Passenger ship and Cargo ship. **Passenger ships** include Ferries and Cruise. A **ferry**is a [merchant vessel](https://en.wikipedia.org/wiki/Merchant_vessel) used to carry passengers, and sometimes vehicles and cargo as well, across a body of water. Ferries are used for short distance. A **cruise ship** or cruise liner is a [passenger ship](https://en.wikipedia.org/wiki/Passenger_ship) used for pleasure voyages, where the voyage itself and the ship's amenities are a part of the experience. A **cargo ship** or freighter is any sort of [ship](https://en.wikipedia.org/wiki/Ship) or vessel that carries [cargo](https://en.wikipedia.org/wiki/Cargo), goods, and materials from one port to another.

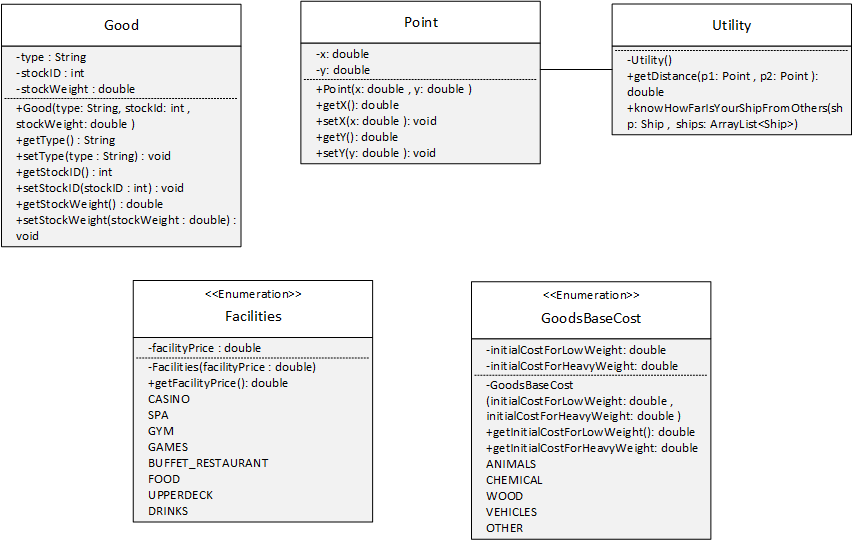


**Fig1: FERRY Fig2: CRUISE**



**Fig3: CARGO SHIP**





Create a **package** and name it as **ship** and create the below described class in this package.

**Point class:**

* Assume variable x as latitude and variable y as longitude.
* **Constructor:** The two-argument constructor is used to initialize the instance variables. Order must be same as mentioned in the UML diagram.
* **Methods:**
  + **getter** and **setter** methods for all the declared instance variables.

Create a **package** and name it as **util.**

**Utility class:**

* **Constructor:** The no-argument private constructor does nothing.
* getDistance(Point p1, Point p2): This static method returns the distance to be travelled by a ship.
* knowHowFarIsYourShipFromOthers(Ship shp, ArrayList<Ship> ships): This static method lets you know how far is your ship from other ships of type String. It can let you know the distance between a Cruise and a Ferry, Ferry and a Cargo ship, Cargo ship and Cruise and vice-versa. The source points act as location of the ship. For Example,

Distance between Stena Line Ferry and

Carnival Breeze Cruise: 5.00

BC Ferries Ferry:32.02

Cargo Altmark Cargo:18.03

**HINT:** You should import the **util** package in order to use these methods in other classes.

Create a **package** and name it as **enums.**

**Facilities:** This Facilities enum is associated with the Passenger Ship class. Based on the facilities chose, the cost will be calculated either for a cruise or a ferry.

Table 1: Facilities

|  |  |
| --- | --- |
| **For Cruise class** | |
| **Facilities (Constants)** | **Value** |
| **CASINO** | **100** |
| **SPA** | **65** |
| **GYM** | **50** |
| **GAMES** | **80** |
| **BUFFETRESTAURANT** | **120** |
| **For Ferry class** | |
| **Facilities (Constants)** | **Value** |
| **FOOD** | **20** |
| **UPPERDECK** | **20** |
| **DRINKS** | **10** |

* **Constructor:** The one-argument constructor is used to initialize the instance variable.
* **Methods:**
  + **getter** methods for the declared instance variable.



**GoodsBaseCost:** The GoodsBaseCost enum is associated with Cargo Ship class. Based on the type of the goods the ship is carrying, the cost will be calculated. Please refer to the table under **CargoShip** class to know when to use initialCostForLowWeight and initialCostForHeavyWeight.

Table 2: Goods Base Cost

|  |  |  |
| --- | --- | --- |
| **Type(Constant)** | **initialCostForLowWeight** | **initialCostForHeavyWeight** |
| ANIMALS | 1000 | 2000 |
| CHEMICAL | 1000 | 3000 |
| WOOD | 900 | 2000 |
| VEHICLES | 1100 | 2500 |
| OTHERS | 800 | 1600 |

* **Constructor:** The two-argument constructor is used to initialize all the instance variables. Arguments must be in same order as per UML.
* **Methods:**
  + **getter** methods for all the declared instance variables

Define the below classes in **ship** package

**Ship class:**

* **Constructor:** The no-argument constructor does nothing.
* **Constructor:** The six-argument constructor is used to initialize all the instance variables. Arguments must be in same order as per UML.
* **Methods:**
  + **getter** and **setter** methods for all the declared instance variables
  + **toString():** This method overrides the **toString()** of **Object** class. The returned String contains manufacturer name.

**PassengerShip class:**

* **Constructor:** The no-argument constructor calls the super class no argument constructor.
* **Constructor:** The seven-argument constructor initializes all the instance variables of super class and also the instance variables of the **PassengerShip** class which are given in the arguments. Order must be same as mentioned in UML diagram. Also, this constructor must initialize **facilities**.
* **Methods:**
  + **getter** and **setter** methods for all the declared instance variables.
  + **addFacilities(String options):**This method returns an ArrayList of facilities of type String. This method is used to add facility options taken from a String to an ArrayList**.** Use comma as identifier and use split function to extract the options and add the facilities to the list.

For Example,

Consider options as a string from the file,

Options = Casino,Spa,Gym

Output: ArrayList<Facilities>

* + **calculateCost():** This method returns the basePrice of type double.
  + **toString():** This method overrides the **toString()** of **Ship** class. The returned String contains facilities.

**Cruise Class:**

* **Constructor:** The no-argument constructor which calls super class no argument constructor.
* **Constructor:** The seven-argument constructor is used to initialize all the instance variables of the super class. The order must be same as mentioned in the UML diagram.
* **Methods:**
  + **getter** and **setter** methods for all the declared instance variables.
  + **calculateCost():** This method overrides the calculateCost() of **PassengerShip** class. This method calculates the cost based upon the facilities which will be read from the input file. Facilities is an enum which has constant values. Please refer to **enums** package to know the values. The returned value will be the sum of baseprice and the total cost calculated for the facilities.
* **toString():** This method should return the cruise name, model, model series, base price, facilities from super class and the cost calculated above of type String in the below format.

cruiseName: Caribbean Princess, BZ09925670

The Base price of the ticket is: $110.0

The facilities chosen are [Casino, Spa, Gym, Games] and the total cost calculated is: $405.0

**Ferry Class:**

* **Constructor:** The no-argument constructor which calls super class no argument constructor.
* **Constructor:** The eight-argument constructor is used to initialize all its instance variables and the instance variables of the super class. Order must be same as mentioned in the UML diagram.
* **Methods:**
  + **getter** and **setter** methods for all the declared instance variables.
  + **getSpeed():** This method returns the speed of the ferry of type double. Speed can be calculated based on the distance.
  + **calculateCost():**This method overrides the calculateCost() of **PassengerShip** class. This method calculates the cost based upon the facilities which will be read from the input file. Facilities is an enum which has constant values. Please refer to the **enums** package to know the values. The returned value will be the sum of basePrice and the total cost calculated for the facilities.
  + **toString():**This method should return the ferry name, model, model series, base price, facilities from super class, the cost, and speed of the ferry, of type String in the below format.

Ferry Name: Stena Line Ferry, FX02925670

The Base price of ticket is: $80.0

The facilities chosen are [Food, Upperdeck] and the total cost is: $120.0

The Speed at which the ferry is travelling: 15.37mph

**CargoShip Class:**

* **Constructor:** The no-argument constructor which calls the super class no argument constructor.
* **Constructor:** The six-argument constructor is used to initialize the instance variables of the super class. Order must be same as mentioned in the UML diagram. Also, this constructor must initialize **goods**.
* **Methods**
  + **addGoods(Good options):** This method returns an ArrayList of options of type Good. This method is used to add **goods** options taken from a String to an ArrayList**.**
* **getDistance():** This method returns the distance between source and destination of type double.
* **calcCostBasedOnWeightNDist():** This method returns the cost based on the stock weight and distance, of type double. Refer the below table to calculate the cost. The InitialCostForHeavyWeight and InitialCostForLowWeight must be retrieved from the enum **GoodsBaseCost** based on the type of the goods.

**Table 3: Cost to be calculated based on stock weight and distance**

|  |  |  |
| --- | --- | --- |
| **Stock Weight** | **Distance** | **Cost** |
| greater than or equal to 2000 | greater than or equal to 1000 | sum of InitialCostForHeavyWeight and 2.60 times the distance |
| less than 2000 | greater than or equal to 1000 | sum of InitialCostForLowWeight and 2.60 times the distance |
| less than 2000 | less than 1000 | sum of InitialCostForLowWeight and 1.33 times the distance |
| greater than or equal to 2000 | less than 1000 | sum of InitialCostForHeavyWeight and 1.33 times the distance |

* **calculateCostWithTax():** This method applies tax on the value returned from **calcCostBasedOnWeightNDist()** and returns the final cost needs to be payable to the ship, of type double. The tax is applied based on the type of stock. Refer the below table to know the tax rates.

**Table 4: Tax rates**

|  |  |
| --- | --- |
| **Type** | **Tax** |
| Chemical | 14% |
| Wood | 10% |
| Food | 12% |
| Other | 8% |

* **toString():** This method returns the total cost after the tax is calculated in the below format.

The total cost calculated to carry the stock is: $7414.35

**Good Class:**

* **Constructor:** The three-argument constructor is used to initialize the instance variables. Order must be same as mentioned in the UML diagram.
* **Methods:**
  + **getter** and **setter** methods for all the declared instance variables.

**Javadoc:**

Include Javadoc comments, using the @author, @param, and @return annotations when appropriate. Generate documentation for your project by clicking on Run from the NetBeans menu bar and then selecting Generate Javadoc. The documentation will be placed in a javadoc subfolder of the dist subfolder inside your project folder. To view the documentation, open the index.html file that is created.

**ShipDriver Class:**

You are given freedom to put in your own logic for this class. Use necessary Conditional and Structural concepts here. Make sure your output exactly matches with the sample output provided to you. Do not hardcode anywhere, we will use different input file to test your project. If you feel, you missed somewhere with your own logic, please refer the points below to continue.

* The input text file is provided to you, name it as ship.txt. This file should be placed in your project folder.
* This class has a method **main** and uses one **Scanner** object to read input from the file.
* Please look at the sample output and print the necessary statements in the same format.
* Look at the sample output and read the input values from the file appropriately.
* Declare a list of shipList of type Ship and initialize it to ArrayList<>(). This list is used to store the details of all the ships given in the input file.
* Declare and create a Cruise object and pass the read values to its parameters if the input read is a Passenger Ship of model Cruise.
* Read the facilities from the file and pass them to addFacilities(String options) method.
* Print the Cruise toString() method using Cruise object.
* Now find how far is the cruise from other ships knowHowFarIsYourShipFromOthers(Ship shp, ArrayList<Ship> ships). Look at the sample output to know the format.
* If the Passenger Ship model is a Ferry, pass the read values to the Ferry class. Do not instantiate the object of Ferry with Ferry, instead come up with an alternative strategy. Also, write a single line comment about how your strategy works.
* Read the facilities from the file and pass them to addFacilities(String options) method.
* Print the toString() method of Ferry class.
* Now find how far is the ferry from other ships knowHowFarIsYourShipFromOthers(Ship shp, ArrayList<Ship> ships). Look at the sample output to know the format.
* If it is a Cargo Ship, create an object for the Good class and pass the appropriate values to it. Also, create an object for the CargoShip class and pass the appropriate values.
* Pass the Good object to addGoods(Good options) .
* Now get the distance to be travelled by the ship and print it. Look at the sample output for the format.
* Print the type of stock carried by the ship and its weight.
* Print the toString() method of CargoShip.
* Now find how far is the cargo ship from other ships knowHowFarIsYourShipFromOthers(Ship shp, ArrayList<Ship> ships). Look at the sample output to know the format.

**Sample Output:**

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Welcome to Kaiser Shipyards  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Cruise details:  Cruise Name: Carnival Breeze Cruise, VX12225670  The Base price of the ticket is: $120.0  The facilities chosen are: [Casino, Spa, Gym, Games, BuffetRestaurant] and the total cost calculated is: $535.0  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ferry details:  Ferry Name: BC Ferries Ferry, FR14525670  The Base price of ticket is: $80.0  The facilities chosen are [Food, Upperdeck, Drinks] and the total cost is: $130.0  The Speed at which the ferry is travelling: 6.39mph  Distance between BC Ferries Ferry and  Carnival Breeze Cruise:56.57  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Cargo Ship details:  Cargo Ship Name: Cargo Altmark Cargo  The distance to be travelled by the ship is 1787.11 miles  The type of stock on the ship is 'Animals' weighing 1500.0 tons  The total cost calculated to carry the stock is: $6098.21  Distance between Cargo Altmark Cargo and  Carnival Breeze Cruise:22.36  BC Ferries Ferry:78.10  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ferry details:  Ferry Name: Stena Line Ferry, FX02925670  The Base price of ticket is: $80.0  The facilities chosen are [Food, Upperdeck] and the total cost is: $120.0  The Speed at which the ferry is travelling: 15.37mph  Distance between Stena Line Ferry and  Carnival Breeze Cruise: 5.00  BC Ferries Ferry:60.21  Cargo Altmark Cargo:18.03  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Cruise details:  Cruise Name: Caribbean Princess Cruise, BZ09925670  The Base price of the ticket is: $110.0  The facilities chosen are: [Casino, Spa, Gym, Games] and the total cost calculated is: $405.0  Distance between Caribbean Princess Cruise and  Carnival Breeze Cruise:42.92  BC Ferries Ferry:13.65  Cargo Altmark Cargo:64.55  Stena Line Ferry:46.61  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Cargo Ship details:  Cargo Ship Name: Kaiser Lines Cargo  The distance to be travelled by the ship is 800.48 miles  The type of stock on the ship is 'Chemical' weighing 6000.0 tons  The total cost calculated to carry the stock is: $4633.69  Distance between Kaiser Lines Cargo and  Carnival Breeze Cruise:551.54  BC Ferries Ferry:499.80  Cargo Altmark Cargo:568.59  Stena Line Ferry:553.47  Caribbean Princess Cruise:512.28  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Cruise details:  Cruise Name: Carnival Destiny Cruise, CD91125670  The Base price of the ticket is: $100.0  The facilities chosen are: [Casino] and the total cost calculated is: $200.0  Distance between Carnival Destiny Cruise and  Carnival Breeze Cruise:40.56  BC Ferries Ferry:37.80  Cargo Altmark Cargo:61.72  Stena Line Ferry:45.55  Caribbean Princess Cruise:29.73  Kaiser Lines Cargo:535.34  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Cargo Ship details:  Cargo Ship Name: SS Shipping Cargo  The distance to be travelled by the ship is 901.33 miles  The type of stock on the ship is 'Wood' weighing 1900.0 tons  The total cost calculated to carry the stock is: $2308.65  Distance between SS Shipping Cargo and  Carnival Breeze Cruise:14.14  BC Ferries Ferry:42.43  Cargo Altmark Cargo:36.06  Stena Line Ferry:18.03  Caribbean Princess Cruise:28.78  Kaiser Lines Cargo:538.52  Carnival Destiny Cruise:31.48  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Cruise details:  Cruise Name: Pacific Rim Cruise, CZ08125670  The Base price of the ticket is: $110.0  The facilities chosen are: [Casino, Games] and the total cost calculated is: $290.0  Distance between Pacific Rim Cruise and  Carnival Breeze Cruise:56.44  BC Ferries Ferry:18.15  Cargo Altmark Cargo:78.80  Stena Line Ferry:60.95  Caribbean Princess Cruise:20.62  Kaiser Lines Cargo:510.36  Carnival Destiny Cruise:25.08  SS Shipping Cargo:43.26  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ferry details:  Ferry Name: DC Ferry, FR14525670  The Base price of ticket is: $80.0  The facilities chosen are [Food, Upperdeck, Drinks] and the total cost is: $130.0  The Speed at which the ferry is travelling: 26.57mph  Distance between DC Ferry and  Carnival Breeze Cruise:28.28  BC Ferries Ferry:28.28  Cargo Altmark Cargo:50.00  Stena Line Ferry:32.02  Caribbean Princess Cruise:14.64  Kaiser Lines Cargo:525.55  Carnival Destiny Cruise:27.15  SS Shipping Cargo:14.14  Pacific Rim Cruise:30.94  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

**Submit your solution by following the steps below:**

* Save your files in **NetBeans**.
* Zip your entire Project. (It should be called ***Lastname*\_Lab07 InheritanceAndPolymorphism**.zip where the **Lastname** is your last name.)
* Submit the Zip file to the **Lab07 InheritanceAndPolymorphism** dropbox.
* Download the Zip file you have submitted.
* Look in the Zip file and verify the class files, javadocs in the Zip folder are updated. If not, resave your project in **NetBeans** and resubmit.