ArrayList Program

You have been asked to develop a menu-driven system for the Luxury Ocean Cruise Outings company. This company coordinates cruise bookings for luxury ship owners and cruise lines. Luxury Ocean Cruise Outings has requested a simple menu-based system to perform core functions that are essential to its daily operations. In order to achieve this task, you will need four classes: Ship, Passenger, Cruise and Driver.

# Ship Class:

Instance variables:

* Ship Name
* Number of each type of room: balcony, ocean view, suite, interior
* A Boolean variable to indicate whether it is in service

Methods:

* Constructor that aceepts all the instance variables
* Getters and setters for all variables
* printShipData method – this method will be similar to what you would do in a toString but we are going to use the toString for something else, so here we will output all the data about the ship. See the sample output.
* toString method – this method will only return the name of the ship

# Passenger Class:

Instance variables:

* name of the passenger
* name of the cruise the passenger is scheduled for
* room type the passenger has requested

Methods:

* Default constructor
* Custom constructor that accepts all three instance variables
* Getters and setters for all instance variables
* printPassenger method – this method will output the details for a single passenger
* toString – this method will only return the name of the passenger

# Cruise Class:

Instance variables

* name of the cruise
* name of the ship
* departure port
* destination
* return port

Methods:

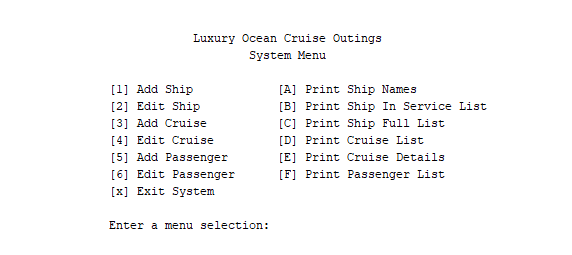
* default constructor
* custom constructor with all instance variables passed in
* Getters and setters for all instance variables
* printCruiseDetails – This method will output the details for a single cruise
* toString – this method will return the cruise name

# Driver Class:

* Create three array lists to store a Ship, a Cruise, and a Passenger. Make these lists global.

Methods:

* initializeShipList – this method will create four ships and load them into the ship array list. All the information will be hard coded in. This will give you data to start with to test your other methods with, without having to load a bunch of data each tme.
* initializeCruiseList – this method will create three cruises and load them into the cruise array list
* initializePassengerList – this method will create six passengers and load them into the passenger array list
* printShipList- this method accepts a String that indicates the type of list you want to be printed. It will then output information about the ship based on the type that was passed in.
  + If there are no ships in the list, you should print “There are no ships to print.”.
  + If the type is “name”, it will print a list of the names of all the ships using the toString method of ship
  + If the type is active, it will print a list of all the ships that are currently active, meaning it is in service. An active ships means it has been assigned a cruise
  + If the type is inactive, it will print a list of all the ships that are currently inactive, meaning in service is false. An inactive ship means that it has not been assigned a cruise.
  + If the type is full, it will print a list of all the ships in the list.
* printCruiseList – this method accepts a String that indicates the type of list you want to be printed. It will then output information about the cruise based on the type that was passed in.
  + If there are no cruises on the list, you should print “There are no cruises to print.”
  + If the type is “list”, it will print only the name of all cruises.
  + If the type is “details”, it will print all of the details of the cruise
* printPassengerList – this method outputs the details of each passenger
* displayMenu – this method will display the menu options. The menu will look like this:



* addShip – this method will add a ship to the ship array list. It must first prompt the user for information about the ship.
* editShip – this method will edit information about a ship already in the list. It will prompt the user to indicate which ship. Your program should display a menu that displays all the ships on the list. The user will then choose which ship they wish to edit. It should then display a menu of the ship information to edit. The user would then choose that option and make the requested change. One of the options on the menu should be update all items.
* addCruise – this method will add a cruise to the cruise array list. It must prompt the user for information about the cruise. You cannot add a ship to a cruise unless the cruise is in the ship list. You can only add a ship that is not in service with another cruise
* editCruise – this method will edit information about a cruise already in the list. The same format should be applied as with ship. Make sure when you edit data that you are enforcing add restrictions
* addPassenger - this method adds a new passenger. It must first prompt the user for information about the passenger. You cannot add a cruise to the passenger list unless it is on the cruise list and in service
* editPassenger – this method works the same as the other two edit methods. Make sure when you edit data that you are enforcing add restrictions

Grading Criteria:

Ship Class- 10 points

Passenger Class – 10 points

Cruise Class – 10 points

Driver Class (70 points)

* three initialize methods – 4 points each
* printShipList- works for any valid argument passed – 5 points
* printCruiseList – works for any valid argument passed – 5 points
* printPassengerList – 5 points
* displayMenu- 3 points
* three add methods – 5 points each
* three edit methods – 5 points each
* Program navigation moves from screen to screen seamlessly – 5 points
* Correct menus are dynamically displayed times – 5 points

Failure to Compile – deduct 60 points  
Failure to Run – deduct 30 points

Missing internal comments – deduct 5 points

Missing Javadoc comments – deduct 5 points

Unprofessional output – deduct 3 points

Late – Deduct 10 points for first week

Deduct 25 points after first week