

IT 145 Project Three Guide

Use this step-by-step guide to complete all the methods needed for the software application's menu system.

TIP: As you work on your code, you can add sample data to test your code changes. Remember that every time you exit the program, your sample data will be cleared. To avoid this issue, it is recommended that you hard-code your sample data as illustrated in the beginning of the Driver.java file. Some hardcoded data has already been created for you, but you may add more if you would like. Look for these in-line comments at the start of the code:

```
// hardcoded ship data for testing
// Initialize ship list
```

Part One: Complete the printCruiseDetails() Method

1. Open the Cruise.java file. Pay attention to the variable names and the code for the constructors, accessors, and mutators, which have already been created by your development team. See below for where you will add code to finish the printCruiseDetails() method.

```
// print cruise details
public void printCruiseDetails() {

    // complete this method

}
```

2. Next, review the required functionality of the "Print Cruise Details" menu option below.

Menu Option	Functionality
Print Cruise Details	<ul style="list-style-type: none">• Prints a list, each cruise on a separate row• Data should be placed in columns that correspond to the variables: cruise name, cruise ship name, departure port, destination, and return port

3. **Print Cruise Details Method:** Write code for the `printCruiseDetails()` method. When your code is complete, test the output by running the `Driver.java` class. Be sure the completed method does the following:
 - a. Prints console output of all cruises and their details or, if there are no cruises in the inventory, the output states this to the user
 - b. Prints the output in an easy-to-read format for the end users (for example, easy-to-read output would present data in aligned columns)

TIP: Within the `Ship.java` file, there is a `printShipData()` method that is very similar to the method you are tasked with creating. You can refer to that method as an example.

4. **Industry Standard Best Practices:** To ensure clarity, consistency, and efficiency among all software developers, your code must do the following:
 - a. Follow appropriate naming conventions for variables and methods
 - b. Include descriptive in-line comments for all code you create and modify

Part Two: Complete the `printShipList()` Method

1. Open the `Driver.java` file. See below for where you will add code to finish the `printShipList()` method.

```
} else if (listType == "active") {  
    System.out.println("\n\nSHIP LIST - Active");  
  
    // complete this code block
```

2. Next, review the required functionality of the “Print Ship List” menu option below.

Menu Option	Validation Check(s)	Functionality
Print Ship In Service List	If no ships are in the inventory, informs the user	Prints a list consisting of all ship names that are in service

3. **Print Ship List Method:** Complete the code for the `printShipList()` method. When your code is complete, test the output. Be sure that the completed method provides console output that returns ship details based on the String value passed to the method.

TIP: The `printShipList()` method already supports printing ship names and a full ship list. You can review the code as you edit the `printShipList()` to support printing ships in service. See below for an example of the system's printed output for the full ship list. Actual results do not need to match this example exactly. A [text version](#) of this table is also available.

SHIP LIST - Full					
SHIP NAME	Number of Rooms				In Service
	Bal	OV	Ste	Int	
Candy Cane	20	40	10	60	true
Peppermint Stick	10	20	5	40	true
Bon Bon	12	18	2	24	false
Candy Corn	12	18	2	24	false

4. **Industry Standard Best Practices:** To ensure clarity, consistency, and efficiency among all software developers, your code must do the following:
 - a. Follow appropriate naming conventions for variables and methods
 - b. Include descriptive in-line comments for all code you create and modify

Part Three: Complete the `addShip()` Method

1. In the `Driver.java` file, see below for where you will add code to finish the `addShip()` method.

```
// Add a New Ship
public static void addShip() {

    // complete this method

}
```

2. Next, review the required functionality of the “Add Ship” menu option below.

Menu Option	Validation Check(s)	Functionality
Add Ship	<ul style="list-style-type: none">- Ensures ship does not already exist in our system- Ensures all class variables are populated	Adds ship to system

3. **Add Ship Method:** Write the code for the addShip() method. When your code is complete, test the output. Be sure that the completed method does the following:

- a. Adds a new Ship object
- b. Includes all class variables
- c. Updates appropriate ArrayList

TIP: You can refer to the Ship.java class constructor to make sure you have included all variables.

4. **Industry Standard Best Practices:** To ensure clarity, consistency, and efficiency among all software developers, your code must do the following:
- a. Ensure that all user input is validated with appropriate feedback to the user
 - b. Include exception handling
 - c. Follow appropriate naming conventions for variables and methods
 - d. Include descriptive in-line comments for all code you create and modify

Part Four: Complete the addCruise() Method

1. In the Driver.java file, see below for where you will add code to finish the addCruise() method.

```
// Adda New Cruise
public static void addCruise() {

    // complete this method

}
```

2. Next, review the required functionality of the “Add Cruise” menu option below.

Menu Option	Validation Check(s)	Functionality
Add Cruise	<ul style="list-style-type: none">- Ensures cruise does not already exist in our system- Ensures all class variables are populated	Adds cruise to system

3. **Add Cruise Method:** Write the code for the addCruise() method. When your code is complete, test the output. Be sure that the completed method does the following:
 - a. Prompts the user for input
 - b. Requires all class variables when creating a cruise
 - c. Validates the ship name and ensures it is in service
 - d. Ensures ship name is not already assigned a cruise
 - e. Adds the new cruise to the cruiseList ArrayList if validation checks pass
4. **Industry Standard Best Practices:** To ensure clarity, consistency, and efficiency among all software developers, your code must do the following:
 - a. Ensure that all user input is validated with appropriate feedback to the user
 - b. Include exception handling
 - c. Follow appropriate naming conventions for variables and methods
 - d. Include descriptive in-line comments for all code you create and modify

Part Five: Complete the main() Method

1. At the beginning of the Driver.java file, see below for where you will add code to finish the main() method.

```
public static void main(String[] args) {  
  
    initializeShipList();           // initial ships  
    initializeCruiseList();        // initial cruises  
    initializePassengerList();     // initial passengers  
  
    // add loop and code here that accepts and validates user input  
    // and takes the appropriate action. include appropriate  
    // user feedback and redisplay the menu as needed
```

2. **Main Method:** Complete the code for the main() method. When your code is complete, test the output. Be sure that the completed method does the following:
- Includes a loop that allows the user to interact with the menu until they signal that they want to exit the system
 - Maps user input to the specified functionality based on the all menu options listed below. A [text version](#) of this table is also available.

Luxury Ocean Cruise Outings
System Menu

[1] Add Ship	[A] Print Ship Names
[2] Edit Ship	[B] Print Ship In Service List
[3] Add Cruise	[C] Print Ship Full List
[4] Edit Cruise	[D] Print Cruise List
[5] Add Passenger	[E] Print Cruise Details
[6] Edit Passenger	[F] Print Passenger List
[x] Exit System	

Enter a menu selection:

Note: The methods for Edit Ship, Edit Cruise, and Edit Passenger do not need to be completed but should still be mapped. When you test your menu, if you enter 2, 4, or 6 for a menu selection, you should get a response that the “feature is not yet implemented.”

3. **Industry Standard Best Practices:** To ensure clarity, consistency, and efficiency among all software developers, your code must do the following:
- Ensure that all user input is validated with appropriate feedback to the user
 - Follow appropriate naming conventions for variables and methods
 - Include descriptive in-line comments for all code you create and modify

Part Six: Submission

Upon completing your code for each class file, review the **What to Submit** section of the Project Three overview and submit all files to the Project Three submission.