

# Assignment 4

2/27/2022

30/30 Points

Offline Score:  
**30/30**

Add Comment

0

## ▼ Details

All assignments are emailed to [cislabs05@gmail.com](mailto:cislabs05@gmail.com)

This assignment has two exercises.

### Exercise 1

#### Part 1

Create a class `SavingsAccount`. Use a static class variable to store the `annualInterestRate` for each of the savers. Each object of the class contains a private instance variable `savingsBalance` indicating the amount the saver currently has on deposit. Provide method `calculateMonthlyInterest` to calculate the monthly interest by multiplying the balance by `annualInterestRate` divided by 12; this interest should be added to `savingsBalance`. Provide a static method `modifyInterestRate` that sets the `annualInterestRate` to a new value. Write a driver program to test the class `SavingsAccount`. Instantiate two different `savingsAccount` objects, `saver1` and `saver2`, with balances of \$2000.00 and \$3000.00, respectively. Set `annualInterestRate` to 4%, then calculate the monthly interest and print the new balances for each of the savers. Then set the `annualInterestRate` to 5% and calculate the next months interest and print the new balances for each of the savers.

#### Part 2

Write another class `SpecialSavings` that extends `SavingsAccount` to pay interest of 10% on accounts that have balances that exceed 10K. Also provided methods to deposit and take money out of savings account. Write a driver program to test the class `SpecialSavings`. Instantiate two different `savingsAccount` objects, `saver1` and `saver2`, with balances of \$2000.00 and \$3000.00, respectively. Make a few deposits and withdrawals and show balance and interest earned for each account.

### Some Tips for helping you with Lab 4:

For Lab 4 part 1 -

Pl. follow the instructions in the assignment and write the class definition. A driver program is the code you would write in `main()` to exercise the code of class definition.

You should declare private variable in your class definition. You should also write protected methods to

<https://deanza.instructure.com/courses/23660/modules/items/1808043><https://deanza.instructure.com/courses/23660/modules/items/1808043>

You need to write deposit and withdrawal methods. Should you put this in parent class or child class? Try to answer this question so that you have most reusability in your class definition.

In this part - you have to learn how polymorphism works.

You will have accounts whose balance might be above or below 10K. By using methods in both classes try to change the interest earned to 10% if balance is above 10K or 4% if the interest is lower.

## Exercise 2

Design a Ship, CargoShip and CruiseShip class being mindful of behavior of each. Demonstrate the classes in a program that has a Ship array. Assign various Ships, CruiseShip and CargoShip to the array elements.

CargoShip and CruiseShip are a child of Ship class.

Pl. submit a class diagram for each exercise, along with code and test runs.

✓ **View Rubric**



(<https://deanza.instructure.com/courses/23660/modules/items/1808043>)

(<https://deanza.instructure.com/courses/23660/modules/items/1808043>)

Rubric of Exercise 1, Exercise 2			
Criteria	Ratings		Pts
Description of criterion No errors, program always works correctly and meets the specification(s). <a href="#">view longer description</a>	<b>2 pts</b> <b>Full Marks</b>	<b>0 pts</b> <b>No Marks</b>	/ 2 pts
Code could be reused as a whole or each routine could be reused. <a href="#">view longer description</a>	<b>2 pts</b> <b>Full Marks</b>	<b>0 pts</b> <b>No Marks</b>	/ 2 pts
Concepts of Inheritance and Polymorphism are applied. <a href="#">view longer description</a>	<b>7 pts</b> <b>Full Marks</b>	<b>0 pts</b> <b>No Marks</b>	/ 7 pts
Java coding conventions are followed. <a href="#">view longer description</a>	<b>1 pts</b> <b>Full Marks</b>	<b>0 pts</b> <b>No Marks</b>	/ 1 pts
Code Readability (as suggested in class). <a href="#">view longer description</a>	<b>1 pts</b> <b>Full Marks</b>	<b>0 pts</b> <b>No Marks</b>	/ 1 pts
Adequately tested (unique test cases, covering boundary conditions). <a href="#">view longer description</a>	<b>1 pts</b> <b>Full Marks</b>	<b>0 pts</b> <b>No Marks</b>	/ 1 pts
Class diagram is provided (UML usage is not	<b>1 pts</b> <b>Full Marks</b>	<b>0 pts</b> <b>No Marks</b>	/ 1 pts


<https://deanza.instructure.com/courses/23660/modules/items/1808043>
<https://deanza.instructure.com/courses/23660/modules/items/1808043>

Rubric of Exercise 1, Exercise 2		
Criteria	Ratings	Pts
		Total Points: 0

<

(<https://deanza.instructure.com/courses/23660/modules/items/1808043>)

(<https://deanza.instructure.com/courses/23660/modules/items/1808043>)