Setting Up A Hierarchical Data Structure in Existing Module Code
5004 Object Oriented Design

In Module 6, we learned about hierarchical representations of data and how to abstract them using generic tree nodes. In this lab, you will be asked to add on three functionalities to Module 6's starter code.

1. **Goals:**

    - Demonstrate you can examine existing code
    - Extend the functionality of an existing complex program
    - Walk through a module's code
    - Approach class design from a different angle

2. **In Recitation:**

At this point you all have experience setting up and creating projects. In small groups (assigned by your TAs), prepare a synthesized list of steps to walk through in examining new code and determining how to create the new requested functionality for that code. Make sure that these steps address the following questions:

- When you are reading someone else's code and you find a keyword or package you are unfamiliar with implementing, what helps you best understand how to use it?
- What do you feel is most often missing from online tutorials or the documentation available through Oracle?
- What is the point of understanding how introductory java utility packages are crafted before implementing those packages and their methods?

Objective 1 : Share your experiences

In your individual groups discuss the following and remember your objective is to learn and to help your fellow students. In addressing the above questions, create your guide for working with new coding concepts and turn them in to your ICE as a PDF.

Objective 2 : Strategy session

Review all of the tasks below for this week's lab assignment and strategize with your group. Review any of the code from the module you don't understand and discuss with each other. For example, maybe it's helpful to look up what a bifunction is again!

What do you think is going to be the hardest and what is going to be the easiest?

Objective 3 : Create your plan

Take a few minutes to brainstorm how to establish the hierarchy in your starter code. There are some proposed employees to add, but none of them have been implemented. If you need help thinking of what to do, ask your team!

3. Instructions:

Open the StarterCode zip file from Canvas inside your IDE and inspect the code with your team. You are welcome to discuss with one another what you notice about the code and what you think might be missing from it.

There is a driver file with an addEmployee method that currently does not work. You will need to undo the comment block around it and design a hierarchy using the different employee classes.

There are also three tasks inside the driver comments that request you add more functionality to the code. Take note of these tasks and work toward implementing them in the code.

4. Reflection:

Each assignment must include a short reflection. The generation of it should take you no more than 15 minutes. This gives you a chance to reflect back on what you learned. For this reflection, please answer the following questions:

1. What did you learn that you were not expecting to in doing this project?
2. How was tree traversal in this program different from the BST example shared in class on July 3?

5. Submission:

Please read carefully. Failure to follow submission instructions can result in a reduced score.

Submit all files on Canvas under the appropriate assignment. Make sure to include the following:

Submit your files as a single zip file named: `"Your Name"_"Assignment".zip`

Your zip file should contain all files needed to make the application run. Include your reflection as: `"lab_7_reflection.pdf"`

Submission checklist:

- [ ] Did you include adequate comments?
- [ ] Did you include comment blocks at the top of each file?
- [ ] Did you name your files as requested?
- [ ] Does your code compile?
- [ ] Did you take care of any warnings presented by your IDE?