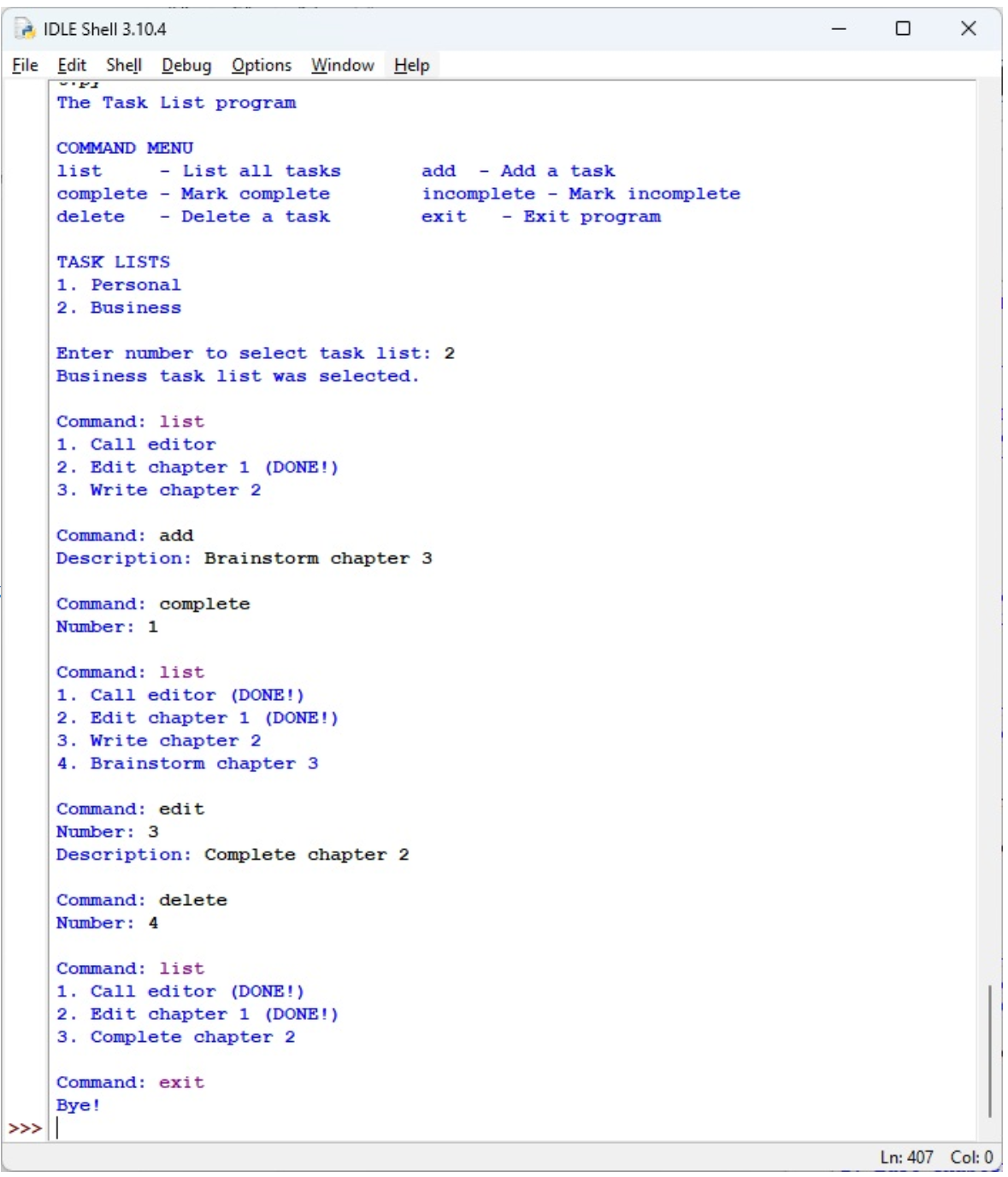
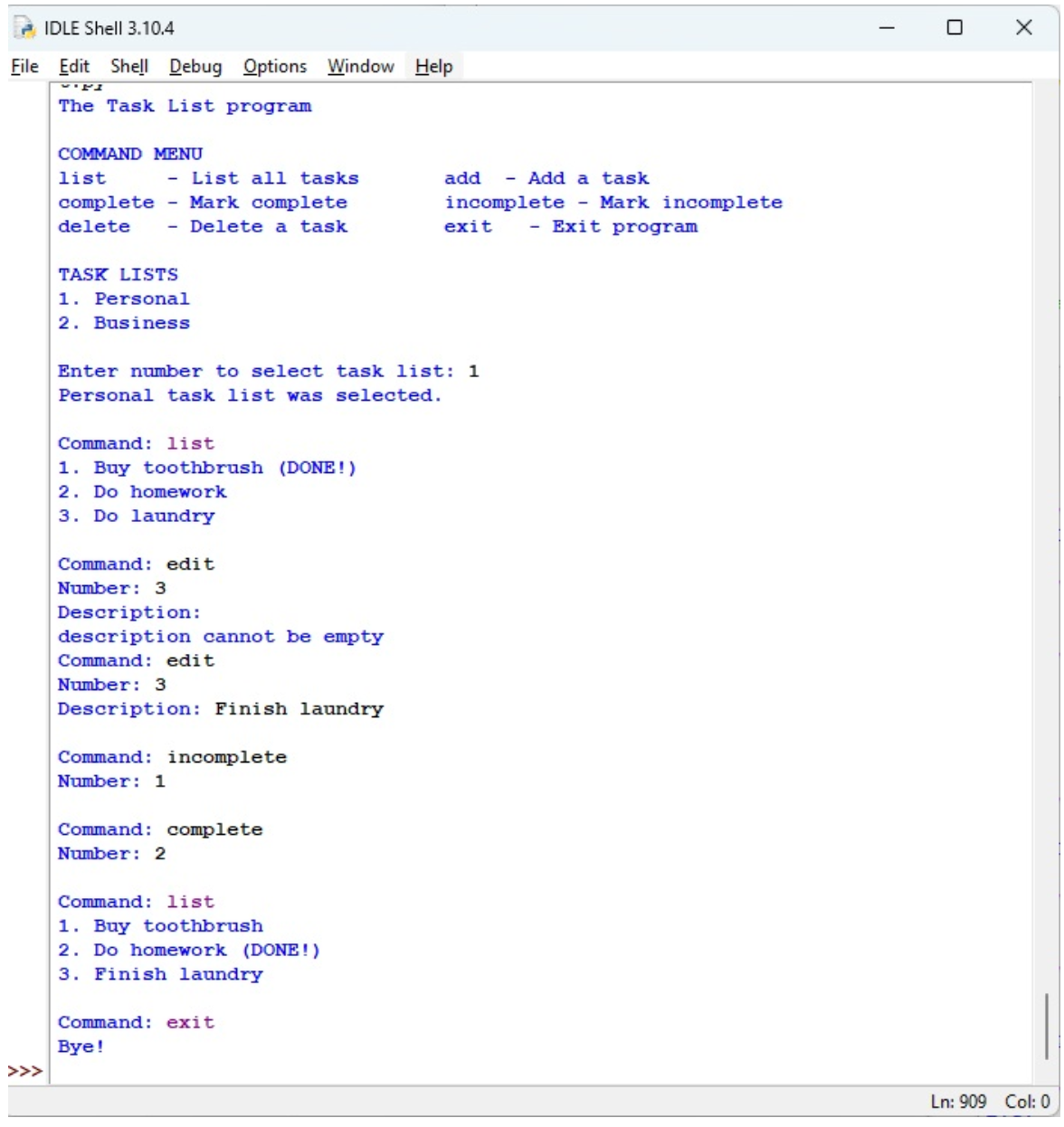
**Design and Implement Business Tier Classes**

In this assignment, you’ll design and implement the business objects for a program that stores one or more task lists. The program is able to work with one or more different task lists and each task list can have zero or more tasks.

You should design classes that will support operations shown in the below screenshots of a program that could use your business tier objects (***Note: You will not be implementing the presentation tier shown here, only the business tier classes):***

The UI should be able to work with multiple task lists. Once a task list is selected, it should be able to list tasks, add a task, delete a task. It also should be able to mark a task complete or incomplete. It should be able to edit the description of a task.

**Part 1: Design the business objects for the program**

1. Identify the data attributes.
2. Subdivide each attribute into its smallest useful components.
3. Identify the classes by sorting the data attributes into categories. (Hint: You will need two classes.)
4. Identify the methods and properties by drawing a UML diagram for the classes.
5. Refine the classes, attributes, methods, and properties. When you’re done, you should be ready to start coding and testing the business objects for the program.

Submit a word document describing the classes and their design. Include the UML diagram (similar to the ones shown in the textbook). You can simply sketch the diagram on paper and include a picture of it, or use a text editor to draw the diagram. It doesn't have to be fancy. It should clearly convey the classes and their attributes, methods and properties. Mark if any are read-only properties clearly (again as shown in Figure 16-5 on page 451 in the textbook).

**Part 2: Implement the business tier classes and test them**

1. Implement the business classes based on the UML diagram.
2. Test the business classes by writing a main function in that file that
   1. creates a test task list
   2. adds a few tasks
   3. lists the tasks in the list
   4. marks one of the tasks complete.
   5. edits the name of one of the tasks
   6. marks a task incomplete
   7. attempts to set the description to empty, but fails.

 As you work on these steps, you can further refine the classes, attributes, methods, and properties for the business classes. Submit your implementation in a file with a name of the form first\_last\_task\_list\_objects.py