

## BCIS 3630

### Object Oriented Programming

#### Problem #4: Data Validation, Multiple (Internal and External) Methods and Objects

This problem is an extension of Problem #3. You are to continue to use the same input and output configuration as for the previous problem. However, you are to employ the currency format for all dollar amounts displayed as output (rather than using a literal \$ symbol in your output.)

For this problem, you are to enhance the performance of the application by performing the following validations:

- No field is permitted to be empty. Even if you are using default values, make sure the user doesn't clear the field and then press "OK" on the prompting screen.

- Renter/lodger name should be at least 4 characters long.

- Occupants (party size) should be numeric and between 1 and 8, inclusive.

- Rental unit name should be at least 8 characters long.

- Arrival date should be exactly 10 characters long.

- Duration of stay should be numeric and between 1 and 14 days, inclusive.

- Per-night rental cost should be numeric and between \$50 and \$350.

Each validation that fails should be accompanied by an appropriate error message pane that (a) identifies the specific field (by name) that caused the error to occur, (b) the specific type of error encountered and (c) the actual value entered into the field. The "error" pane should display a non-default icon.

#### Additional Requirements:

This program should utilize the following:

- At least one internal method (other than "main," within the main file containing the main method)

- At least one external method (outside the file containing the main method) that is not part of or imbedded in an object.

- At least one object creation and direct utilization as an active part of your coded solution. (This should be an object of your making, not one created from one of the standard Java packages.)

You are to supply standard elements (hardcopy, softcopy, sample output generated, etc.) for this problem. You should also supply hardcopy of the class diagram for this application. Show all the classes you created and connectivity to other classes in your application.

Your "startup" code should be executable from a file designated as "**Prob4App**".

**Solutions developed and directly executable in NetBeans will earn extra credit (5 points). Partially imported applications will receive partial extra credit.**

**Due Date: October 29<sup>th</sup>**