

**CSIS 2175****Due date: April 11, 2022 (5:00PM)****Submission**

You need to submit **the Entire Eclipse project** in a single .zip file by the due date. **NO LATE SUBMISSION** will be allowed.

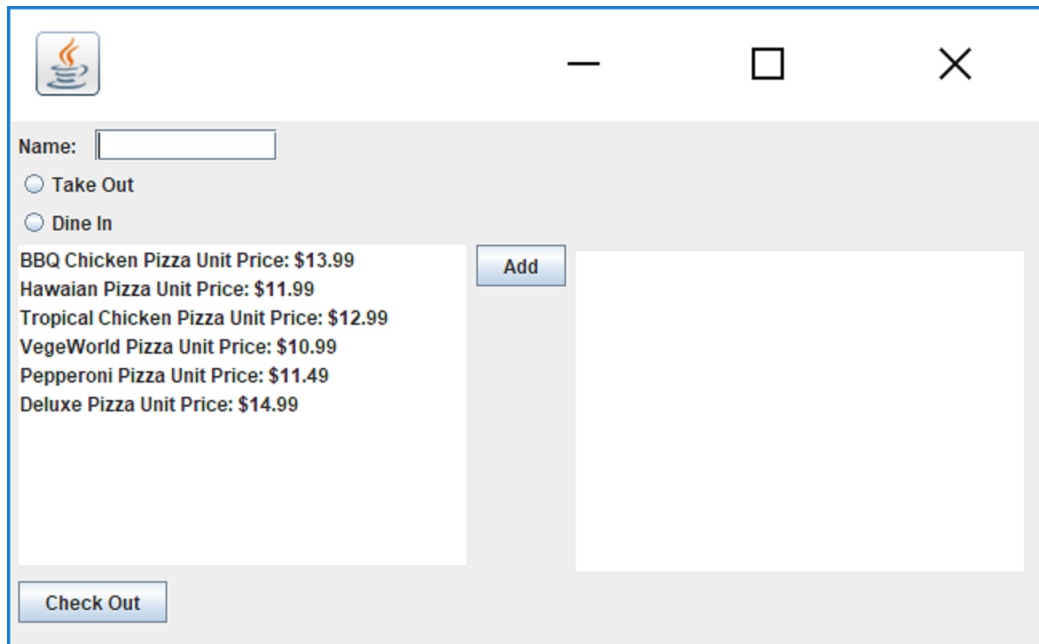
You may submit your work multiple times, but only the last submission will be graded.

**Description**

In this assignment, you are required to implement an electronic pizza ordering system. User can order pizzas via this system.

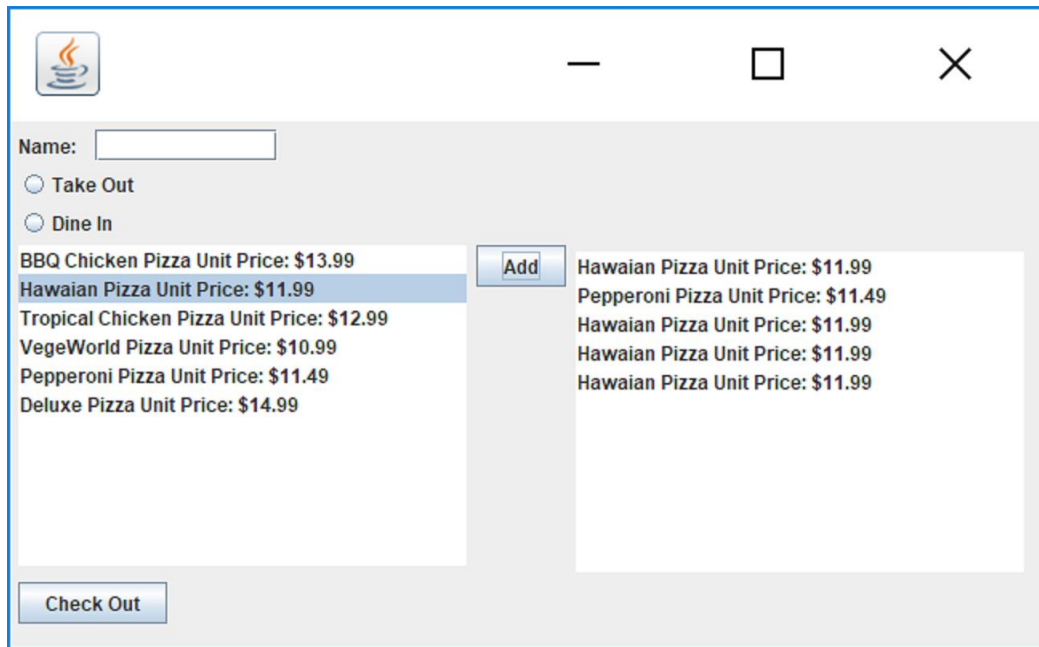
**Your Task**

You are asked to create a GUI as follows:



When running the program, it loads the choices of pizzas from the file menu.acddb and displays in the listbox on the left (Menu List). (\*The database contains a table called menu, which has three fields, id (int), pizza (String), price (double))

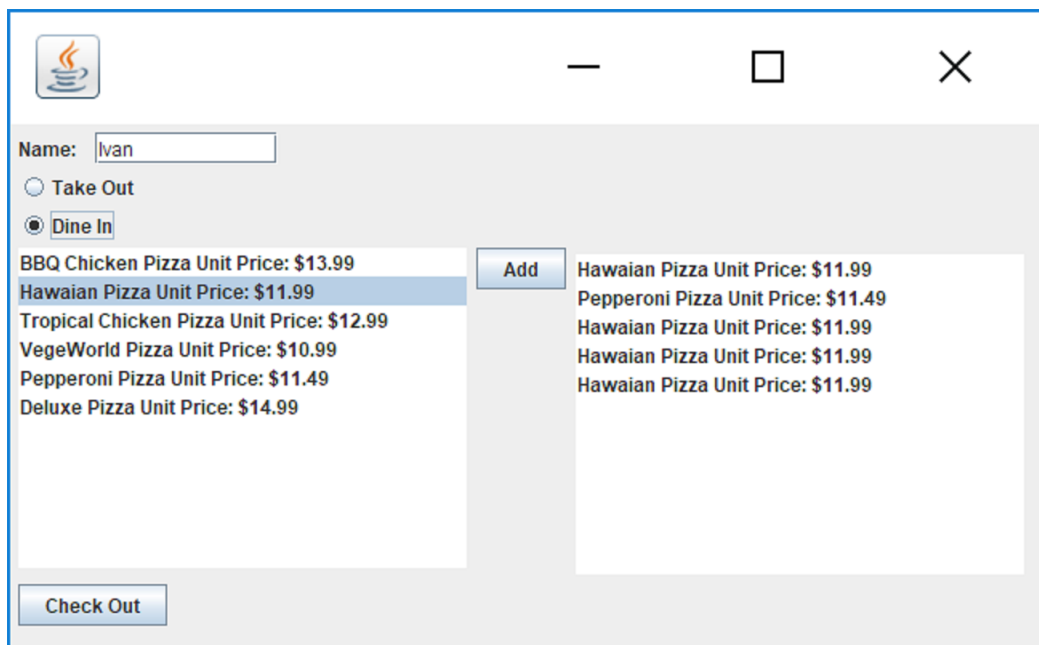
User can click the button "Add" to add pizza to the listbox on the right (Order List). The Order List will accumulate the pizzas added and may contain repeated items. For example, the following figure shows five added pizzas in the order: Hawaiian Pizza, Pepperoni Pizza, Hawaiian Pizza, Hawaiian Pizza, Hawaiian Pizza:



The screenshot shows a Java Swing window with a title bar containing a logo, a minus button, a maximize button, and a close button. The window contains the following elements:

- Name:** A text field with an empty input.
- Service Type:** Two radio buttons, "Take Out" (selected) and "Dine In".
- Pizza List (Left):** A list of pizza options with their unit prices:
  - BBQ Chicken Pizza Unit Price: \$13.99
  - Hawaian Pizza Unit Price: \$11.99 (highlighted)
  - Tropical Chicken Pizza Unit Price: \$12.99
  - VegeWorld Pizza Unit Price: \$10.99
  - Pepperoni Pizza Unit Price: \$11.49
  - Deluxe Pizza Unit Price: \$14.99
- Add Button:** A button labeled "Add" located between the two lists.
- Pizza List (Right):** A list of the currently selected items:
  - Hawaian Pizza Unit Price: \$11.99
  - Pepperoni Pizza Unit Price: \$11.49
  - Hawaian Pizza Unit Price: \$11.99
  - Hawaian Pizza Unit Price: \$11.99
  - Hawaian Pizza Unit Price: \$11.99
- Check Out Button:** A button labeled "Check Out" at the bottom left.

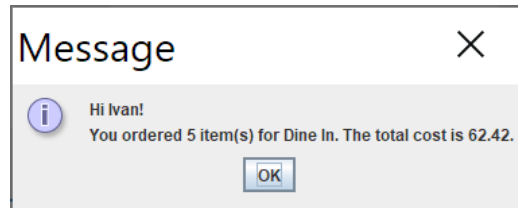
User can input his/her name in the Name textfield, and choose only one of the radio buttons, Take Out or Dine In. The following figure shows a sample input:



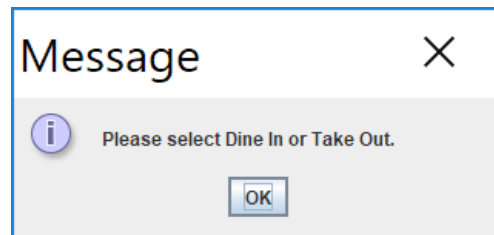
This screenshot shows the same Java Swing window as the previous one, but with sample input:

- Name:** The text field now contains the name "Ivan".
- Service Type:** The "Dine In" radio button is now selected.
- Pizza List (Left):** The same list of pizza options as before.
- Add Button:** The "Add" button remains between the lists.
- Pizza List (Right):** The list of selected items remains the same as in the previous screenshot.
- Check Out Button:** The "Check Out" button remains at the bottom left.

After that, user can click the button "Check Out" and the program shows the following message:



The total cost is calculated by summing up all the unit price of those ordered pizzas + 5% tax in the order. If the user does not check the radio buttons, it shows the following message:



### Requirement and Hints:

1. The layout of the GUI should be as close to the figures above as possible.
2. You must create a class called Pizza to store a pizza record.
3. The following code can return a String given a floating number, say 12.34567, in two decimal point:

```
String str = String.format("%.2f", 12.34567); // str will be "12.35"
```

### Grading

Correctness of the program: 90%

Programming style/comment/clarity: 10%

**Overall marks will not be more than 50% if not following the requirement.**

### Assumptions

You may assume that there will be no invalid input by the user.