In-Class Exercise: Party Picker

# Concepts:

* Inheritance, GUI with JOptionPane, ArrayLists & For-Each loops

# Instructions

* Create a GUI program to enter the names of potential guests to a party, the food they usually bring, and whether or not they’re actually invited.
* Your program should have a superclass of Person (first name, last name, isInvited) and a subclass of Friend (foodToBring).
* Create proper constructors for each class, passing values via super() as required.
* The Person superclass should have a getter for isInvited, as well as a getFullName() method, which will return the concatenated full name of the person, with a space between.
* The Friend subclass should have a toString() method which builds a string describing the person, their food and whether they are invited or not. It should call the super’s getFullName() method and include some basic logic to use the super’s isInvited to build an appropriate message about each potential guest. The method should return something like:
  + “Jane Doe is bringing pizza. They are invited to the party.” **OR**
  + “John Doe is bringing broccoli. They are NOT invited to the party.”
* Your Main() class should run the program, doing the following:
  + Create an ArrayList that will hold all Friend objects entered by the user.
  + Display a welcome/instructions popup message when the program starts
  + Continue to ask the user to enter names and foods until they decide to stop.
  + When each potential guest’s data is entered, create and save a new Friend object.
  + After all guests are entered, use a for-each loop to retrieve each guest from the list and build a report to list each guest, their food and whether they are invited.

# Exercise Notes

* Don’t worry about input validation unless you want to.
* For the time being, you can ignore all Cancel buttons in the popups.
* This exercise has you build classes using inheritance (Person, Friend). It may seem a bit over-engineered to do so, since there’s only one subclass… why not just put all the properties/methods in the Person class? A) We need lots of practice with inheritance, and B) Building it this way allows much greater flexibility to our overall design. If we wanted to expand this program later, to include other types of Person we might invite to our parties (Co-Worker, Family, etc.), it’s far easier. Inherit from Person and you’re practically done.







