PGR107 – Python Programming
Spring 2023
Objects and Classes

1. We want to add a button to the tally counter in previous lecture that allows an operator to undo an accidental button click. Provide a method

   def undo (self)

   that simulates such a button. As an added precaution, make sure that the operator cannot click the undo button more often than the count button.

2. Implement a class **Address**. An address has a house number, a street, an optional apartment number, a city, a state, and a postal code. Define the constructor such that an object can be created in one of two ways: with an apartment number or without. Supply a print method that prints the address with the street and house number (and maybe apartment number) on one line and the city, state, and postal code on the next line. Supply a method **def comesBefore (self, other)** that tests whether this address comes before other when compared by postal code.

3. Implement a class **Car** with the following properties. A car has a certain fuel efficiency (measured in miles/gallon) and a certain amount of fuel in the gas tank. The efficiency is specified in the constructor, and the initial fuel level is 0. Supply a method **drive** that simulates driving the car for a certain distance, reducing the fuel level in the gas tank, and methods **getGasLevel**, to return the current fuel level, and **addGas**, to tank up. Sample usage:

   ```
   myHybrid = Car (50) # 50 miles per gallon
   myHybrid.addGas (20) # Tank 20 gallons
   myHybrid.drive (100) # Drive 100 miles
   print (myHybrid.getGasLevel ()) # Print fuel remaining
   ```

4. Provide a class for authoring a simple letter. In the constructor, supply the names of the sender and the recipient:

   def _ _init_ _(self, letterFrom, letterTo)

   Supply a method

   def addLine(self, line)

   to add a line of text to the body of the letter. Supply a method

   def getText(self)

   that returns the entire text of the letter. The text has the form:

   Dear recipient name:
   blank line
   first line of the body
   second line of the body
   . . .
   last line of the body
   blank line
   Sincerely,
   blank line
   sender name

   Also supply a driver program that prints the following letter.

   Dear John:

   I am sorry we must part.
   I wish you all the best.

   Sincerely,

   Mary

   Construct an object of the Letter class and call addLine twice.

5. Design a **Customer** class to handle a customer loyalty marketing campaign. After accumulating $100 in purchases, the customer receives a $10 discount on the next purchase. Provide methods

   def makePurchase (self, amount)

   def discountReached (self)

   Provide a test program and test a scenario in which a customer has earned a discount and then made over $90, but less than $100 in purchases. This should not result in a second discount. Then add another purchase that results in the second discount.

6. Write a class that simulates a bank account. Customers can deposit and withdraw funds. If sufficient funds are not available for withdrawal, a $10 overdraft penalty is charged. At the end of the month, interest is added to the account. The interest rate can vary each month.