A Windows Environment Visual Basic .Net program is to be written that simulates the operation of a gas pump. At any time during the simulation one should be able to determine, from the pump, the amount remaining in the supply tank from which the gas is being pumped. If a request for gas, in gallons, is less than the amount of gas in the tank, the request should be filled; otherwise only the available amount in the supply tank should be used. Each time the gas is pumped, the total and price of the gallons pumped should be displayed. The amount of gas, in gallons, that was pumped should be subtracted from the amount in the supply tank.

For the simulation assume the pump is randomly idle for 1 to 15 minutes between customers and that a customer randomly requests between 3 and 20 gallons of gas. The tank capacity is 500 gallons and initially should simulate a half-hour time frame. Additionally, for each arrival and request for gas we want to know the idle time before the customer arrived, how many gallons of gas were pumped, and the total price of the transaction. The pump itself must keep track of the price of gas and the amount of gas remaining in the tank. Typically, the price per gallon of gas ranges from \$2.50 to \$3.00.

Two class objects are to be created: Pump and Customer. Pump:

Public members- (1) constructor, gal=500, price=\$2.90; (2) utility function that does not receive any arguments or return any arguments, but prints out the formatted amount in the tank and price; (3) request function has one argument (pump amount) and returns no arguments, but determines how much is left in the tank, prints out this amount, and prints the cost of sale.

Private members- (1) amount in the tank (integer); (2) price (double) Customer:

Public: constructor, calls seed, creates and initializes variable for arrival time randomly (minutes 1-16); creates and initializes variable for gallons used randomly (3-20). Simulation Click Event:

- (1) create a pump object with required initial gallons of gas
- (2) display the values in the initialized pump
- (3) set the elapsed time to 0
- (4) obtain customer arrival time //first arrival
- (5) add the arrival to the elapsed time
- (6) while elapsed time does not exceed the simulation time
 - a. display elapsed time
 - b. obtain customer request for gas
 - c. activate the pump with the request
 - d. obtain a customer arrival time //next arrival
 - e. add the arrival time to elapsed time

End loop

(7) display a message that the simulation is over.

The following objects are required on the form:

- 3 command buttons (simulation, clear (clear text boxes and list box), and exit)
- 3 labels for identification
- 2 labels for date and time
- 1 list box
- 3 text boxes

The zipped **project** (Example: CollegeJoeP4 should be e-mailed to <u>cs375@cs.ua.edu</u> no later than 11:59 pm on the due date.

The following information should appear as comment statements before the main body of the program:

Joe College CS375 Project 3 April 5, 2010

Grading Criteria for the project:

Pump Class: constructor, private data members, and public service methods (12 points)

Customer Class: constructor, private data members, and public service methods (8 points)

Form: name and prefix (2 points)

Objects: Three command buttons (prefix and caption), five labels (prefix), three text boxes (prefix), and one list box (prefix). (15 points)

Code: use of indentation, use of meaningful names, creation of objects, access of appropriate pump

methods, use of comments, and appropriate output (23 points)

Total Points: 60

Sample Output follows on the next pages:









