JAVA PROGRAMMING LANGUAGE
Assignment 5

1. Write a *Person* class that contains the following fields and methods:
   - First Name
   - Last Name
   - A unique ID Number (say, 1001, 1002, etc.). The ID should be assigned by the computer. Use a class variable to keep track of the last ID assigned so that you don't repeat the same number twice.
   - Necessary constructors.
   - Methods to return last name, first name, full name, and ID Number
   - Methods to print last name, first name, and ID Number
   - The *toString()* method that returns a neatly formatted string describing the key attributes of the person

   Write a *PersonTester* class to test your class.

2. You operate several hot dog stands distributed throughout town. Define a class named *HotDogStand* that has an instance variable for the hot dog stand's ID number and an instance variable for how many hot dogs the stand has sold that day.

   Create a constructor that allows a user of the class to initialize both values.

   Create a method named *justSold* that increments by one the number of hot dogs the stand has sold. The idea is that this method will be invoked each time the stand sells a hot dog so that you can track the total number of hot dogs sold by the stand.

   Add another method that returns the number of hot dogs sold.

   Finally, add a static variable that tracks the total number of hot dogs sold by all hot dog stands and a static method that returns the value in this variable.

   Write a main method to test your class with at least three hot dog stands that each sell a variety of hot dogs.

3. In Assignment 3 asked you to create a Pizza class describing different kinds of pizzas. Create a PizzaOrder class that allows up to three pizzas to be saved in an order. In addition to appropriate instance variables and constructors, add the following methods:

   - *public void setNumPizzas(int numPizzas)* —sets the number of pizzas in the order. numPizzas must be between 1 and 3.
   - *public void setPizza1(Pizza pizza)* —sets the first pizza in the order.
   - *public void setPizza2(Pizza pizza)* —sets the second pizza in the order.
   - *public void setPizza3(Pizza pizza)* —sets the third pizza in the order.
   - *public double calcTotal()* —returns the total cost of the order
   - *public int getNumPizzas()* — returns the number of pizzas in the order.
   - *public Pizza getPizza1()* — returns the first pizza in the order or null if pizza1 is not set.
   - *public Pizza getPizza2()* — returns the second pizza in the order or null if pizza2 is not set.
   - *public Pizza getPizza3()* — returns the third pizza in the order or null if pizza3 is not set.

- A copy constructor that takes another *PizzaOrder* object and makes an independent copy of its pizzas. This might be useful if using an old order as a starting point for a new order.

Write a main method to test the new methods. Changing the pizzas in the new order should not change the pizzas in the original order.

For example,

```
Pizza pizza1 =                          // Code to create a large pizza, 1 cheese, 1 ham
Pizza pizza2 =                          // Code to create a medium pizza, 2 cheese, 2 pepperoni
PizzaOrder order1 =                     // Code to create an order
order1.setNumPizzas(2);                 // 2 pizzas in the order
order1.setPizza1(pizza1);               // Set first pizza
order1.setPizza2(pizza2);               // Set second pizza
double total = order1.calcTotal();      // Should be 18+20 = 38
PizzaOrder order2 = new PizzaOrder(order1);          // Use copyconstructor
order2.getPizza1().setNumCheeseToppings(3);          // Change toppings
double total = order2.calcTotal();                   // Should be 22 + 20 = 42
double origTotal = order1.calcTotal();               // Should still be 38
```

Note that the first three lines of code are incomplete. You must complete them as part of the assignment.