

Go Nuts for Donuts

Stack Class

1. Using the basic Stack class, make a Stack of Donuts.

```
public class Stack {
    private int count;
    private Object data[] = new Object [50];
    public Stack () {
        count = 0;
    }
    public void push (Object addMe) {
        data [count] = addMe;
        count++;
    }
    public int size () {
        return count;
    }
    public boolean isFull () {
        return (count == 50);
    }
    public Object pop () {
        count--;
        return data [count];
    }
    public Object peek () {
        return data [count--];
    }
    public boolean isEmpty () {
        return count == 0;
    }
    public void clear () {
        count = 0;
    }
}
```



2. Add in the shuffle method:

```
public void shuffle(){
    String names[]={"Bear Claw", "Boston Cream", "Chocolate Glaze", "Cinnamon Twist",
        "Donut Holes","Double Chocolate", "Eclair", "Glazed", "Jelly Filled",
        "Maple Bar", "Maple Glazed", "Milk", "Old Fashioned", "Plain",
        "Powdered",
        "Red Velvet", "Sprinkled", "Strawberry Glazed"};
    int pts[]={2, 3, 2, 4, 2, 1, 5, 6, 7, 2, 2, 2, 3, 1, 0, 0, 2, 2};
    double costs[]={1, 2.3, 4.5, 1, 0.95, 0.85, 0.95, 0.95, 0.85, 0.85, 1,
        2, 3.3,
        0.85, 0.85, 0.85, 2.3};
    //Randomize the order of the arrays
    for (int i = 0; i < 100; i++){
        int r1 = (int) (Math.random() * names.length);
        int r2 = (int) (Math.random() * names.length);
        //swap names array
        String temp = names[r1];
        names[r1] = names[r2];
        names[r2] = temp;
        //swap points array
        int temp1 = pts[r1];
        pts[r1] = pts[r2];
        pts[r2] = temp1;
        //swap cost array
        double temp2 = costs[r1];
        costs[r1] = costs[r2];
        costs[r2] = temp2;
    }
    count = 0;
    //TO DO: push all (now in random order) into the Deck
    for (int i = 0; i < names.length; i++) {
        Donut d = new Donut(names[i], pts[i], costs[i]);
        push(d);
    }
}
```