

Complete The Coin Machine

By: Jimmy Pan

- Complete the coin machine class.
 - Output does not need to match.
 - The amount the coins add up to much be correct.
 - The output format must be kept.
 - Please use the given method `changeGiven`

```
public class Main {
    public static void main(String[] args) {
        CoinMachine cm = new CoinMachine(50, 50, 50, 50);

        cm.giveChange(50.00, 49.99);
        cm.giveChange(35.00, 27.99);
        cm.giveChange(32.00, 25.00);
        cm.giveChange(1.23, 0.95);
        cm.giveChange(0.75, 0.55);
        cm.giveChange(33.33, 27.75);
        cm.giveChange(0.92, 0.50);
        cm.inStorage();
    }
}

public class CoinMachine {
    private int quarter;
    private int dime;
    private int nickel;
    private int penny;

    public CoinMachine() {
        this.quarter = 0;
        this.dime = 0;
        this.nickel = 0;
        this.penny = 0;
    }

    public CoinMachine(int quarter, int dime, int nickel, int penny) {
        this.quarter = quarter;
        this.dime = dime;
        this.nickel = nickel;
        this.penny = penny;
    }

    public void giveChange(double payment, double price) {
        if (payment < price) {
            System.out.println("Customer did not pay enough.");
            return;
        }
    }
}
```

```

        // keep count of amount of each coin that is going to be given to the user
        int q = 0;
        int d = 0;
        int n = 0;
        int p = 0;

        // payment - how much the user pays
        // price - their total for the purchase
        /*****
        // TODO: calculate the amount of coins needed

        *****/
    }

    private void changeGiven(int quarter, int dime, int nickel, int penny) {
        System.out.println("Coins used:");
        System.out.printf("%d Quarter(s)%n", quarter);
        System.out.printf("%d Dime(s)%n", dime);
        System.out.printf("%d Nickel(s)%n", nickel);
        System.out.printf("%d Penny(s)%n", penny);
    }

    public void inStorage() {
        System.out.printf("%-10s%d%n", "25¢", this.quarter);
        System.out.printf("%-10s%d%n", "10¢", this.dime);
        System.out.printf("%-10s%d%n", "5¢", this.nickel);
        System.out.printf("%-10s%d%n", "1¢", this.penny);
    }

    public void addQuarter(int count) {
        if (count < 0) {
            System.out.println("Negative not allowed.");
        } else {
            this.quarter = this.quarter + count;
        }
    }

    public void addDime(int count) {
        if (count < 0) {
            System.out.println("Negative not allowed.");
        } else {
            this.dime = this.dime + count;
        }
    }

    public void addNickel(int count) {
        if (count < 0) {
            System.out.println("Negative not allowed.");
        } else {
            this.nickel = this.nickel + count;
        }
    }
}

```

```

    public void addPenny(int count) {
        if (count < 0) {
            System.out.println("Negative not allowed.");
        } else {
            this.penny = this.penny + count;
        }
    }
}

```

Sample Expected Output

```

Coins used:
0 Quarter(s)
0 Dime(s)
0 Nickel(s)
1 Penny(s)
Coins used:
28 Quarter(s)
0 Dime(s)
0 Nickel(s)
1 Penny(s)
Coins used:
22 Quarter(s)
15 Dime(s)
0 Nickel(s)
0 Penny(s)
Coins used:
0 Quarter(s)
2 Dime(s)
1 Nickel(s)
3 Penny(s)
Coins used:
0 Quarter(s)
2 Dime(s)
0 Nickel(s)
0 Penny(s)
Coins used:
0 Quarter(s)
31 Dime(s)
49 Nickel(s)
3 Penny(s)
Coins used:
0 Quarter(s)
0 Dime(s)
0 Nickel(s)
42 Penny(s)
25¢      0
10¢      0
5¢       0
1¢       0

```