Homework Turnin

Email:	rgalanos@fcps.edu
Section:	6G
Course:	TJHSST APCS 2016-17
Assignment:	01-02
Receipt ID:	8c2d41e321f047eb49b06163441697b2

Turnin Successful!

The following file(s) were received:

SmartCard_Driver.java (2649 bytes)

```
 //name:     date:
import java.text.DecimalFormat;
public class SmartCard_Driver
{
   public static void main(String[] args)
   {
      Station downtown = new Station("Downtown", 1);
      Station center = new Station("Center City", 1);
      Station uptown = new Station("Uptown", 2);
      Station suburbia = new Station("Suburb", 4);

      SmartCard jimmy = new SmartCard(20.00); //bought with $20.00
      jimmy.board(center);                    //boarded in zone 1
      jimmy.disembark(suburbia);              //disembark in zone 4
      jimmy.disembark(uptown);                //disembark without having boarded

      //lots more test cases!
   }
}
class SmartCard
{
   double balance;
   int zone;
   boolean boarded;
   String city;

   public SmartCard()
   {
      balance = 0.0;
   }
   public SmartCard(double bal)
   {
      balance = bal;
   }

   public void addMoney(double money)
   {
      balance+=money;
   }
   public double getBalance()
   {
      return balance;
   }
   public boolean isBoarded()
   {
      return boarded;
   }
   public void board(Station s)
   {
      if(boarded)
      {
```

```
            System.out.println("Error, already boarded");
            System.exit(0);
        }
        else if(balance<0.5)
        {
            System.out.println("Error, balance is too low");
            System.exit(0);
        }

        zone = s.getZone();
        city =s.getName();
        boarded = true;
        System.out.println("Boarded at: " + city);
    }
    public double cost(Station s)
    {
        if(boarded==false)
        {
            System.out.println("Error, you do not have a starting destination");
            System.exit(0);
        }
        return Math.abs(s.getZone()-zone)*0.75+0.5;

    }
    public void disembark(Station s)
    {
        if(boarded==false)
        {
            System.out.println("Error, not previously boarded");
            System.exit(0);
        }

        balance = balance - cost(s);
        System.out.println("From " + city + " to " + s.getName() + " costs $" + cost(s)+ ".");
        System.out.println("Balance: $" + balance);
        boarded = false;
    }
}
class Station
{
    String name;
    int zone;

    public Station()
    {
        name = "NO NAME";
        zone = 0;
    }
    public Station(String nm, int zn)
    {
        name = nm;
        zone = zn;
    }

    public String getName()
    {
        return name;
    }
    public int getZone()
    {
        return zone;
    }

    public void setName(String nm)
    {
        name = nm;
    }
    public void setZone(int zn)
    {
        zone = zn;
    }

    public String toString()
    {
        return name + ", Zone " + zone;
    }
}
```