//CreditCard.java

```
package strategyDesignPattern;

import java.util.ArrayList;
public class CreditCard {
	private String id;
	private int totalPaybackPoints;
	private ArrayList<Purchase> listOfPurchases;

	public CreditCard(String id, ArrayList<Purchase> listOfPurchases) {
		super();
		this.id = id;
		this.listOfPurchases = listOfPurchases;
	}

	public int getTotalPaybackPoints() {
		return totalPaybackPoints;
	}

	public void setTotalPaybackPoints(int totalPaybackPoints) {
		this.totalPaybackPoints = totalPaybackPoints;
	}

	public void calculateTotalPaybackPoints()
	{
		for (Purchase p : listOfPurchases)
		{
			this.totalPaybackPoints += p. getPaybackpoints();

		}
	}
	public void display()
	{
		this.calculateTotalPaybackPoints();
		System.out.println("Total pay back points for the crdit card no ." + id
+ " are " + this.getTotalPaybackPoints());
	}
}
```

//Purchase.java

```
package strategyDesignPattern;

import java.util.Date;
import java.util.GregorianCalendar;

public class Purchase {
		private Date pdate;
		private int pAmount;
		private int paybackpoints;
```

```
private PaybackPolicy policy;
public Purchase(Date pdate, int pAmount) {
	super();
	this.pdate = pdate;
	this.pAmount = pAmount;
	this.calculatePolicy();
}
public Date getPdate() {
	return pdate;
}
public void setPdate(Date pdate) {
	this.pdate = pdate;
}
public int getpAmount() {
	return pAmount;
}
public void setpAmount(int pAmount) {
	this.pAmount = pAmount;
}
public int getPaybackpoints() {
	return paybackpoints;
}
public void setPaybackpoints(int paybackpoints) {
	this.paybackpoints = paybackpoints;
}
public PaybackPolicy getPolicy() {
	return policy;
}
public void setPolicy(PaybackPolicy policy) {
	this.policy = policy;
}

public void calculatePolicy()
{
	GregorianCalendar cal = new GregorianCalendar(2016,0,1);
	Date fqpStartDate = cal.getTime();

	cal = new GregorianCalendar(2016,2,31);
	Date fqpEndDate = cal.getTime();

	cal = new GregorianCalendar(2016,3,1);
	Date sqpStartDate = cal.getTime();

	cal = new GregorianCalendar(2016,5,30);
	Date sqpEndDate = cal.getTime();

	cal = new GregorianCalendar(2016,6,1);
	Date tqpStartDate = cal.getTime();

	cal = new GregorianCalendar(2016,8,30);
	Date tqpEndDate = cal.getTime();
```

```
                                        cal = new GregorianCalendar(2016,9,1);
                                        Date foqpStartDate = cal.getTime();

                                        cal = new GregorianCalendar(2016,11,31);
                                        Date foqpEndDate = cal.getTime();

                                        if(this.pdate.after(fqpStartDate) &&
this.pdate.before(fqpEndDate) || this.pdate.equals(fqpStartDate)||this.pdate.before(fqpEndDate))
                                        {
                                                this.setPolicy(new FirstQuarterPurchase());
                                        }
                                        else if(this.pdate.after(sqpStartDate) &&
this.pdate.before(sqpEndDate) || this.pdate.equals(sqpStartDate)||this.pdate.before(sqpEndDate))
                                        {
                                                this.setPolicy(new SecondQuarterPurchase());
                                        }
                                        else if(this.pdate.after(tqpStartDate) &&
this.pdate.before(tqpEndDate) || this.pdate.equals(tqpStartDate)||this.pdate.before(tqpEndDate))
                                        {
                                                this.setPolicy(new ThirdQuarterPurchase());
                                        }
                                        else if(this.pdate.after(foqpStartDate) &&
this.pdate.before(foqpEndDate) || this.pdate.equals(foqpStartDate)||this.pdate.before(foqpEndDate))
                                        {
                                                this.setPolicy(new FourthQuarterPurchase());
                                        }

                                }
                                public int computePaybackPoints()
                                {
                                                int points = 0;
                                                points =
this.getPolicy().calculatePaybackPoints(this.pAmount);
                                                return points;

                                }
                                public void display()
                                {
                                        System.out.println("The purchase amount is " + pAmount + "
on " + pdate);
                                }

        }

        //PaybackPolicy.java

        package strategyDesignPattern;

        public abstract class PaybackPolicy {

                Purchase purchase;
```

```
	public PaybackPolicy() {
		super();
	}
	public PaybackPolicy(Purchase purchase) {
		super();
		this.purchase = purchase;
	}
	public Purchase getPurchase() {
		return purchase;
	}
	public void setPurchase(Purchase purchase) {
		this.purchase = purchase;
	}

	public abstract int calculatePaybackPoints(int pAmount);
}
```

//FirstQuarterPurchase.java

```
package strategyDesignPattern;

public class FirstQuarterPurchase extends PaybackPolicy {

	public  int calculatePaybackPoints(int amount)
	{
		int points = 0 ;
		while(amount != 0 && amount >= 300)
		{
			points++;
			amount = amount - 300 ;
		}
		return points;
	}

}
```

//SecondQuarterPurchase.java

```
package strategyDesignPattern;


public class SecondQuarterPurchase extends PaybackPolicy {

	public  int calculatePaybackPoints(int amount)
	{
		int points = 0 ;
		while(amount != 0 && amount >= 200)
		{
			points++;
			amount = amount - 200 ;
```

```
		}
		return points;
	}

}
```

//ThirdQuarterPurchase.java

```
package strategyDesignPattern;

public class ThirdQuarterPurchase extends PaybackPolicy {

	public  int calculatePaybackPoints(int amount)
	{
		int points = 0 ;
		while(amount != 0 && amount >= 150)
		{
			points++;
			amount = amount - 150 ;
		}
		return points;
	}

}
```

//FourthQuarterPurchase.java

```
package strategyDesignPattern;

public class FourthQuarterPurchase extends PaybackPolicy {

	public  int calculatePaybackPoints(int amount)
	{
		int points = 0 ;
		while(amount != 0 && amount >= 100)
		{
			points++;
			amount = amount - 100 ;
		}
		return points;
	}

}
```

//test1.java

```
package Test;
```

```
import java.util.ArrayList;
import java.util.Date;
import java.util.GregorianCalendar;
import strategyDesignPattern.*;

public class test1 {

	public static void main(String[] args)
	{
		GregorianCalendar cal = new GregorianCalendar(2016,1,23);
		Date first = cal.getTime();

		cal = new GregorianCalendar(2016,6,1);
		Date second = cal.getTime();

		cal = new GregorianCalendar(2016,9,14);
		Date third = cal.getTime();

		cal = new GregorianCalendar(2016,11,25);
		Date fourth = cal.getTime();

		System.out.println("Purchases made : ");

		Purchase p1 = new Purchase(first,20000);
		p1.display();
		Purchase p2 = new Purchase(second,30000);
		p2.display();
		Purchase p3 = new Purchase(third,15000);
		p3.display();
		Purchase p4 = new Purchase(fourth,10000);
		p4.display();

		ArrayList<Purchase> listOfPurchases = new ArrayList<Purchase>();
		listOfPurchases.add(p1);
		listOfPurchases.add(p2);
		listOfPurchases.add(p3);
		listOfPurchases.add(p4);

		CreditCard card = new CreditCard("123456789000",listOfPurchases);
		int points1 = p1.computePaybackPoints();
		p1.setPaybackpoints(points1);

		int points2 = p2.computePaybackPoints();
		p2.setPaybackpoints(points2);

		int points3 = p3.computePaybackPoints();
		p3.setPaybackpoints(points3);

		int points4 = p4.computePaybackPoints();
		p4.setPaybackpoints(points4);

		System.out.println("Points : ");
```

```
                card.display();

        }

}
```

//Output

```
Purchases made :
The purchase amount is 20000 on Tue Feb 23 00:00:00 IST 2016
The purchase amount is 30000 on Fri Jul 01 00:00:00 IST 2016
The purchase amount is 15000 on Fri Oct 14 00:00:00 IST 2016
The purchase amount is 10000 on Sun Dec 25 00:00:00 IST 2016
Points :
Total pay back points for the crdit card no .123456789000 are 516
```