Sachin Bhargava
OO Design, Architecture, Methodologies
Object Adapter

Shape Interface

```
public interface Shape {

      public void display();
      public void undisplay();
      public void setLocation (int x, int y);
      public void getLocation();
      public void setColor(String color);
      public void fill();

}
```

Point Class

```
public class Point implements Shape{
	int xcoord = 0;
	int ycoord = 0;
	String color = "null";


	public void display() {
		System.out.println("I am Displaying my point");
	}

	public void fill() {
		System.out.println("I am filled with " + this.color);
	}

	public void getLocation() {
		System.out.println("(x,y) = " + xcoord + ", " +
ycoord);
	}

	public void setColor(String color) {
		this.color = color;
	}

	public void setLocation(int x, int y) {
		this.xcoord = x;
		this.ycoord = y;
	}

	public void undisplay() {
		System.out.println("I am undisplaying my Point");

	}

}
```

Square Class

```
public class Square implements Shape{

	int xcoord = 0;
	int ycoord = 0;
	String color = "null";


	public void display() {
		System.out.println("I am Displaying my Square");
	}

	public void fill() {
		System.out.println("I am filled with " + this.color);
	}

	public void getLocation() {
		System.out.println("(x_bottom_left,y_bottom_left) = "
+ xcoord + ", " + ycoord);
	}

	public void setColor(String color) {
		this.color = color;
	}

	public void setLocation(int x, int y) {
		this.xcoord = x;
		this.ycoord = y;
	}

	public void undisplay() {
		System.out.println("I am undisplaying my Square");

	}

}
```

Triangle Class

```
public class Triangle implements Shape{

	int xcoord = 0;
	int ycoord = 0;
	String color = "null";


	public void display() {
		System.out.println("I am Displaying my
Triangle");
	}

	public void fill() {
		System.out.println("I am filled with " +
this.color);
	}

	public void getLocation() {
		System.out.println("(x_center,y_center) = " +
xcoord + ", " + ycoord);
	}

	public void setColor(String color) {
		this.color = color;
	}

	public void setLocation(int x, int y) {
		this.xcoord = x;
		this.ycoord = y;
	}

	public void undisplay() {
		System.out.println("I am undisplaying my
Triangle");

	}


}
```

MyCircle Class

```
public class MyCircle {

	int xcoord = 0;
	int ycoord = 0;
	String color = null;

	public void fixLocation(int x, int y){
		this.xcoord = x;
		this.ycoord = y;
	}

	public void findLocation(){
		System.out.println("(x_center, y_center) = ("+ xcoord
+", "+ycoord+")");
	}

	public void displayIt(){
		System.out.println("i am displaying MyCircle");
	}

	public void setItsColor(String color){
		this.color = color;
	}

	public void fillIt(){
		System.out.println("I am filled with " + this.color);
	}

	public void undisplayIt(){
		System.out.println("i have undisplayed MyCircle");
	}

}
```

Circle Class

```
public class Circle implements Shape{

	private MyCircle circ;

	public Circle(){
		circ = new MyCircle();
	}

	public void display() {
		circ.displayIt();
	}

	public void fill() {
		circ.fillIt();
	}

	public void getLocation() {
		circ.findLocation();
	}

	public void setColor(String color) {
		circ.setItsColor(color);
	}

	public void setLocation(int x, int y) {
		circ.fixLocation(x, y);
	}

	public void undisplay() {
		circ.undisplayIt();
	}

}
```

Driver

```
public class Driver {

	public static void main(String[] args) {

		Shape p1 = new Point();
		Shape s1 = new Square();
		Shape t1 = new EquiTriangle();

		p1.display();
		s1.undisplay();
		t1.getLocation();
		System.out.println();
		System.out.println();

		Shape c1 = new Circle();

		c1.display();
		c1.undisplay();
		c1.setLocation(3, 2);
		c1.getLocation();
		c1.setColor("red");
		c1.fill();

	}

}
```