Assignment 2: Design Patterns

The purpose of this assignment is to get you familiar with some design patterns discussed in the lecture on modifiability tactics.
Assignments: (1) Explain the following design patterns:
1. facade pattern
2. bridge pattern
3. mediator pattern
4. proxy pattern
5. factory pattern
(2) Give an illustrating example for one of the above patterns. A good example is given as follows:

////////////////Example///

Bridge Pattern

Bridge pattern helps to decouple abstraction from implementation.
To illustrate how the decoupling of abstraction and implementation using a bridge pattern helps to improve the flexibility in your design, consider the following case.
Supposed you are painting with colored pencils. Your toolkit consists of colored pencils of three different sizes: big, medium and small. Each size comes with 12 different colors. So you need a collection of 36 colored pencils for your painting. See how this case can be simplified with brushes. You only need three brushes of different sizes and 12 pigments of different colors. Now your toolkit is reduced from 36 items to 15 items (12 pigments+3 brushes). Decoupling brushes (abstraction, or interface) from colors (implementation) improves flexibility in making your paintings.

This is an interesting example adapted from:
http://www.cnblogs.com/zhenyulu/articles/67016.html. The following code is used to illustrate the above example.

```
using System;

 abstract class Brush
{
 protected Color c;
 public abstract void Paint();

 public void SetColor(Color c)
{ this.c = c; }
}

 class BigBrush : Brush
{
```

```
public override void Paint()
{ Console.WriteLine("Using big brush and color {0} painting…", c.color); }
}

class SmallBrush : Brush
{
public override void Paint()
{ Console.WriteLine("Using small brush and color {0} painting…", c.color); }
}

class Color
{
public string color;
}

class Red : Color
{
public Red()
{ this.color = "red"; }
}

class Blue : Color
{
public Blue()
{ this.color = "blue"; }
}

class Green : Color
{
public Green()
{ this.color = "green"; }
}

class Client
{
public static void Main()
{
Brush b = new BigBrush();
b.SetColor(new Red());
b.Paint();
b.SetColor(new Blue());
b.Paint();
b.SetColor(new Green());
b.Paint();
```

```
        b = new SmallBrush();
        b.SetColor(new Red());
        b.Paint();
        b.SetColor(new Blue());
        b.Paint();
        b.SetColor(new Green());
        b.Paint();
    }
}
```

////////////////Example///

Notes:

(1) Send the assignment as an attachment to: liqiping1991@gmail.com, with the following email subject: "[SA assignment 2]" + Your student number (e.g. [SA assignment 2] 3120102776);

(2) The document must be delivered in Word or PDF format and named as follows: your student number + "_" + your name + "_" + "assignment 2" (e.g. 3120102776_张三_assignment 2.doc).

(3) This assignment will fall due on Apr. 27, 2015.

Hint: use Wikipedia as your reference, or refer to the book: Design Patterns: Elements of Reusable Object-Oriented Software, which is downloadable from the course FTP.