Q1. The following code implements the Command design pattern (example is taken from Wikipedia).

```
/* The Command interface */
public interface Command {
   void execute();
}

/* The Invoker class */
public class Switch {
   public void storeAndExecute(Command cmd) {
      cmd.execute();
   }
}

/* The Receiver class */
public class Light {

   public void turnOn() {
      System.out.println("The light is on");
   }

   public void turnOff() {
      System.out.println("The light is off");
   }
}

/* The Command for turning on the light - ConcreteCommand #1 */
public class FlipUpCommand implements Command {
   private Light theLight;

   public FlipUpCommand(Light light) {
      this.theLight = light;
   }
   /*  INCOMPLETE  */
}

/* The Command for turning off the light - ConcreteCommand #2 */
public class FlipDownCommand implements Command {
   private Light theLight;

   public FlipDownCommand(Light light) {
      this.theLight = light;
   }
```

```
   /* INCOMPLETE  */
}
```

a) Write (pseudo)code that completes classes `FlipUpCommand` and `FlipDownCommand` **[4pts]**

```
/* The Command for turning on the light - ConcreteCommand #1 */
public class FlipUpCommand implements Command {
   private Light theLight;

   public FlipUpCommand(Light light) {
      this.theLight = light;
   }
   public void execute(){
       this.theLight.turnOn();
}
}
```

```
/* The Command for turning off the light - ConcreteCommand #2 */
public class FlipDownCommand implements Command {
   private Light theLight;

   public FlipDownCommand(Light light) {
      this.theLight = light;
   }
   public void execute(){
       this.theLight.turnOff();
}
}
```

b) Write a test program/class/"main function" that uses the implemented command pattern, that turns the light on, and then off. **[6pts]**

```
public class client{
   public void main(){
   Light light = new Light();

   Command turnOn = new FlipUpCommand(light);
   Command turnOff = new FlipDownCommand(light);

   //turn light on
   setCmd(turnOn);
   exCmd();
```

```
    //turn light off
    setCmd(turnOff);
    exCmd();
    }

    public void setCmd(Command cmd){
        this.cmd = cmd;
    }

    public void exCmd(){
        cmd.execute();
    }
}
```

Q2. In the Lesson 9 lecture videos, we cover four Quality Attributes (QAs) (*Availability, Performance, Security, Testability*). Pick the biggest software project you have ever been involved with -- this could be your senior design, your internship, or course projects, etc. Pick one QA from this list that you think is applicable to the project you chose and read Chapter 4 plus the corresponding chapter for the QA, to study the tactics included in the book to achieve the QA.

For this assignment: a) describe the project you chose briefly, b) explain why the QA you chose is important to the project, and c) write about the things you (or others who you work with) have done, that fall into one of the tactics to address the QA for the project. **[20ps]**

For example, if the QA you picked was Performance, then you should read Chapter 4, plus Chapter 8 from the [BASS] textbook, available on Canvas in the Lesson 9 module. Figure 8.3 shows the Performance Tactics in two categories: "Control Resource Demand" and "Manage Resources". You should then write about what you (or your team) did to "control resource demand" or "manage resources".

a) In my senior design class my sponsor is f5 and we are working on developing a capture the flag software for their website which aims to teach users the basics of cyber security. Our project involves developing new capture the flag challenges for the website as well as working on backend enhancements to provide better performance on the website.
b) The QA I am choosing to discuss is security as the major goal of the project is to educate users on the importance of cybersecurity.
c) Security tactics being addressed

Detect Service of Denial: although not fully implemented the idea is to use a Damn Vulnerable Web Application (DVWA) software to launch dummy attacks on a "website" created in a challenge environment so users can understand the ways a website can prevent a denial of service attack as well as how a denial of service attack is able to accomplish its goals of shutting down a website.

Authenticate users: We currently are using Google's authentication platform for users to login to the website and access features such as challenge creation. Without authentication users on the website are unable to access things like challenge creation which take up resources on the website which could overload the website resources due to too many challenges being created at once.

Encrypt Data: We are currently developing a challenge to teach users about encryption that involves the use of a tool called Steghide which is able to embed files into jpg files. The idea is to have users decrypt text files within jpg files to receive a flag for a code and then submit the flag, completing the challenge. The challenge will teach users how the password being supplied to the function that embeds the file in the images uses that password to encrypt data within random pixels of the image, only modifying the least significant bit and to access the info you need the key.