

COMP 4081
Exam 3
Fall 2013

Name (Last, First): Solutions

Rules:

- No potty breaks.
- Turn off cell phones/devices.
- Closed book, closed note, closed neighbor.
- WEIRD! Do not write on the backs of pages. If you need more pages, ask me for some.

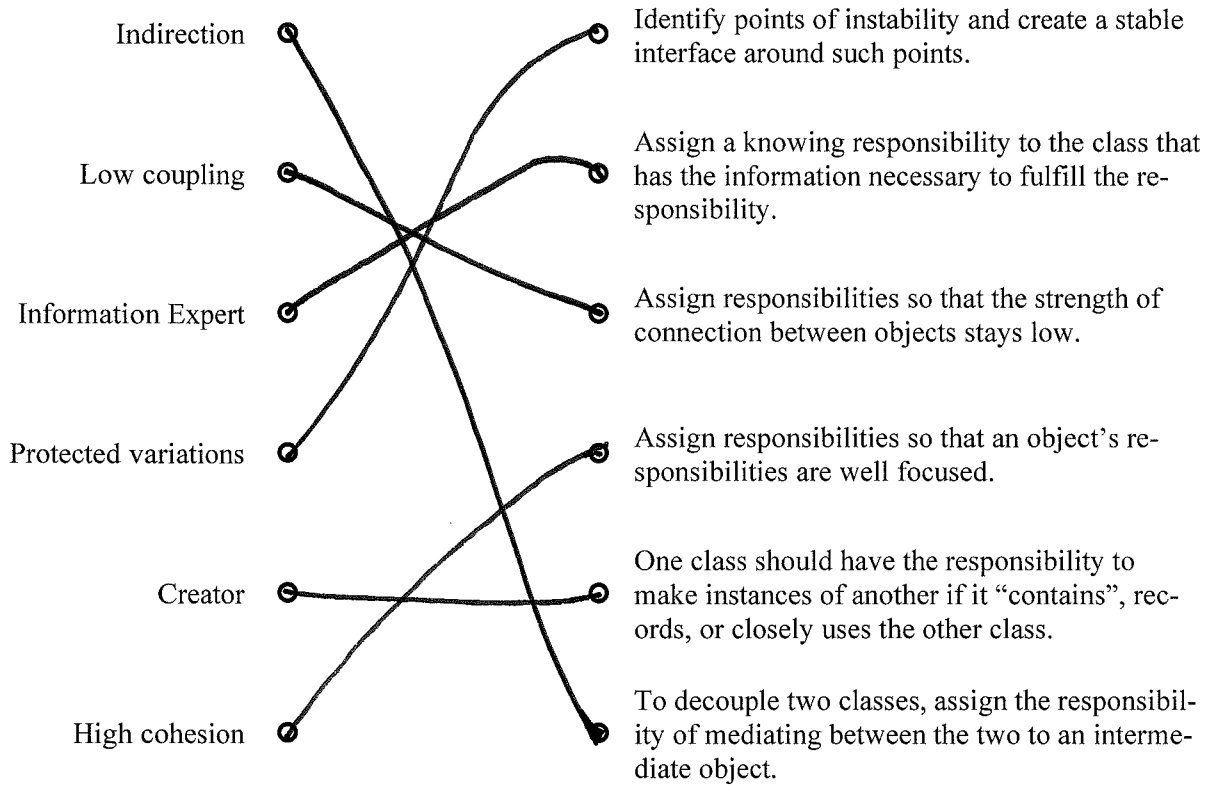
Reminders:

- Verify that you have all pages.
- Don't forget to write your name.
- Read each question carefully.
- Don't forget to answer every question.

Additional Items:

- For questions that involve writing code:
 - You may omit `import` statements.
 - You may omit exception-handling code.

1. [6pts] For each pattern below, draw a line from the pattern to its definition.



2. [2pts] T or F? Coupling and cohesion are closely linked in that as one increases, so does the other.

a. True

b. False

3. [2pts] Given classes A and B , which of the following is not a common type of coupling in Java?

a. A is a direct or an indirect subclass of B

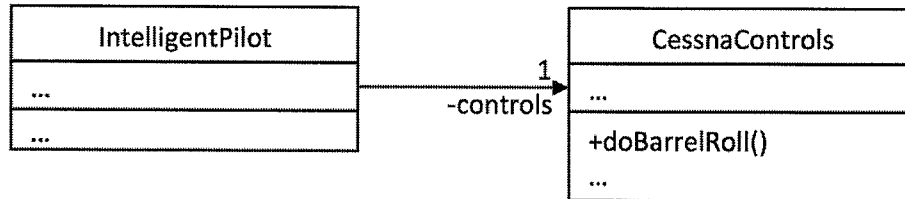
b. A belongs to the same package as B

c. A method parameter or local variable in A references B

d. A has an instance variable that refers to B

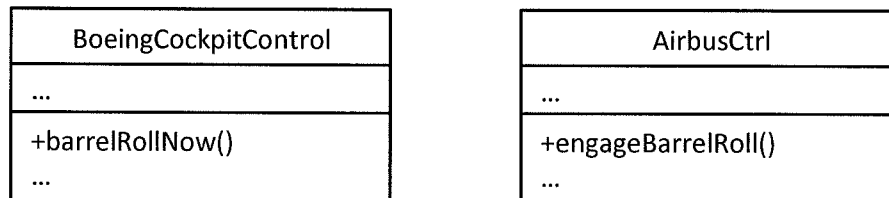
e. A invokes methods of B

4. [10pts] Imagine that you are the creator of an “intelligent” autopilot system that can actually fly and land real airplanes (wow!). Initially, you implemented your system to fly small Cessna airplanes. Here is an excerpt of your current software design:



Note that the Cessna Aircraft Company provided the software interface for controlling the plane (CessnaControls), and you created the intelligent decision-making part (IntelligentPilot).

As your next step, you would like your system to support different types of airplanes other than Cessnas. For example, Boeing and Airbus each provide their own software control interfaces:

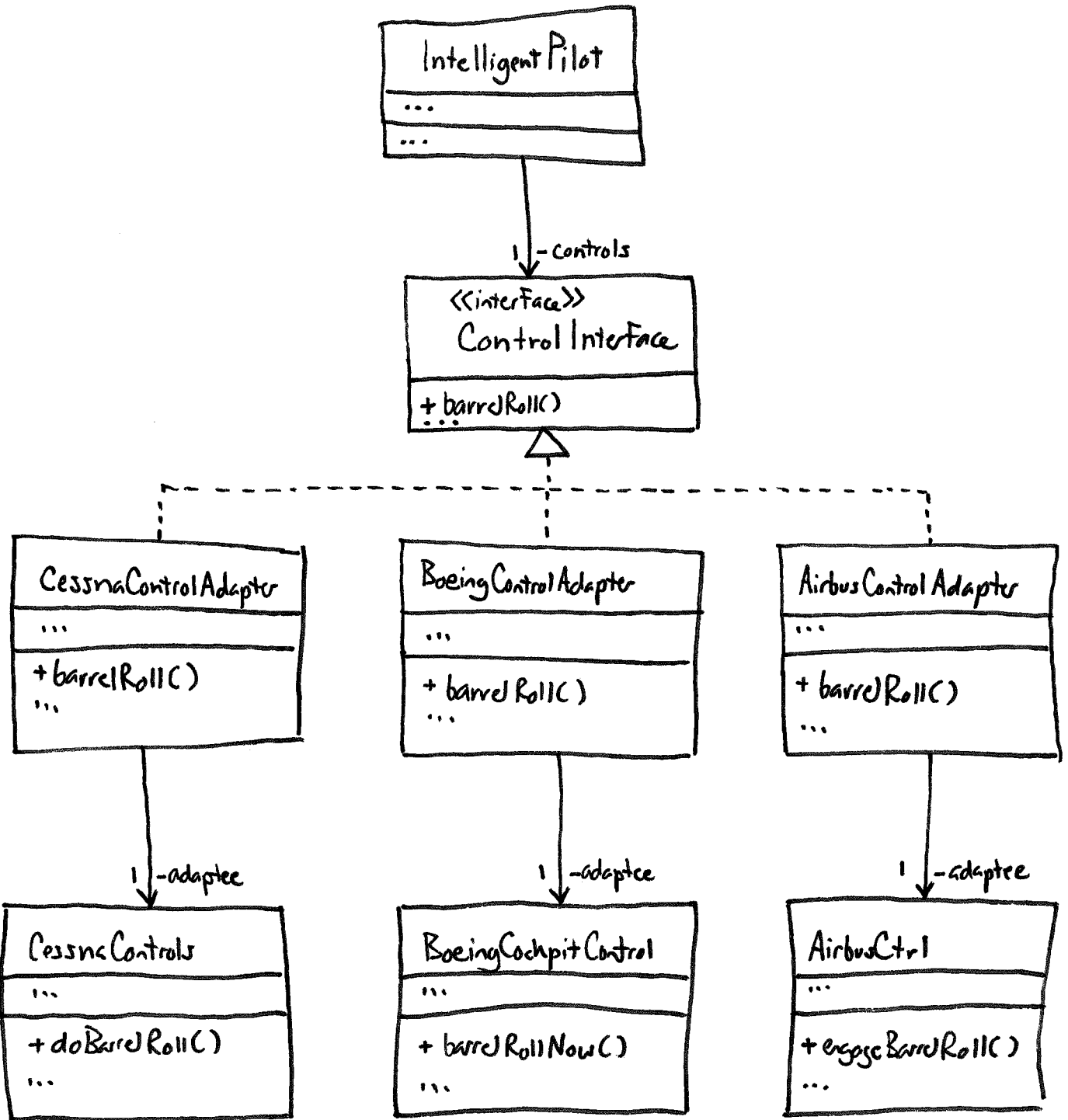


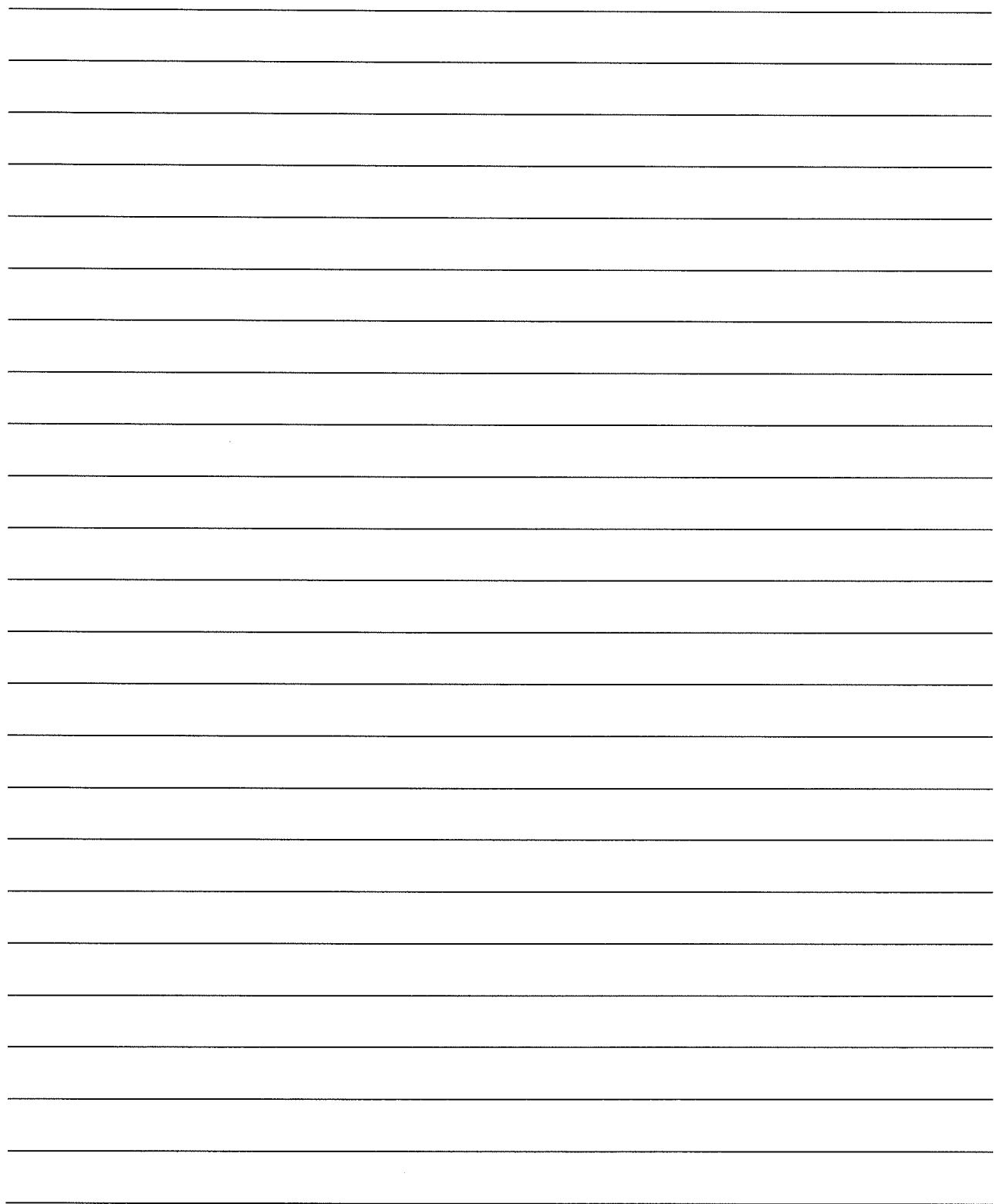
Update your current software design to allow easy switching between control systems. Your design must apply the **Indirection Pattern**, the **Protected Variations Pattern**, and the **Polymorphism Pattern**. (Hint: Recall that we discussed a similar design in class for switching between tax calculators.)

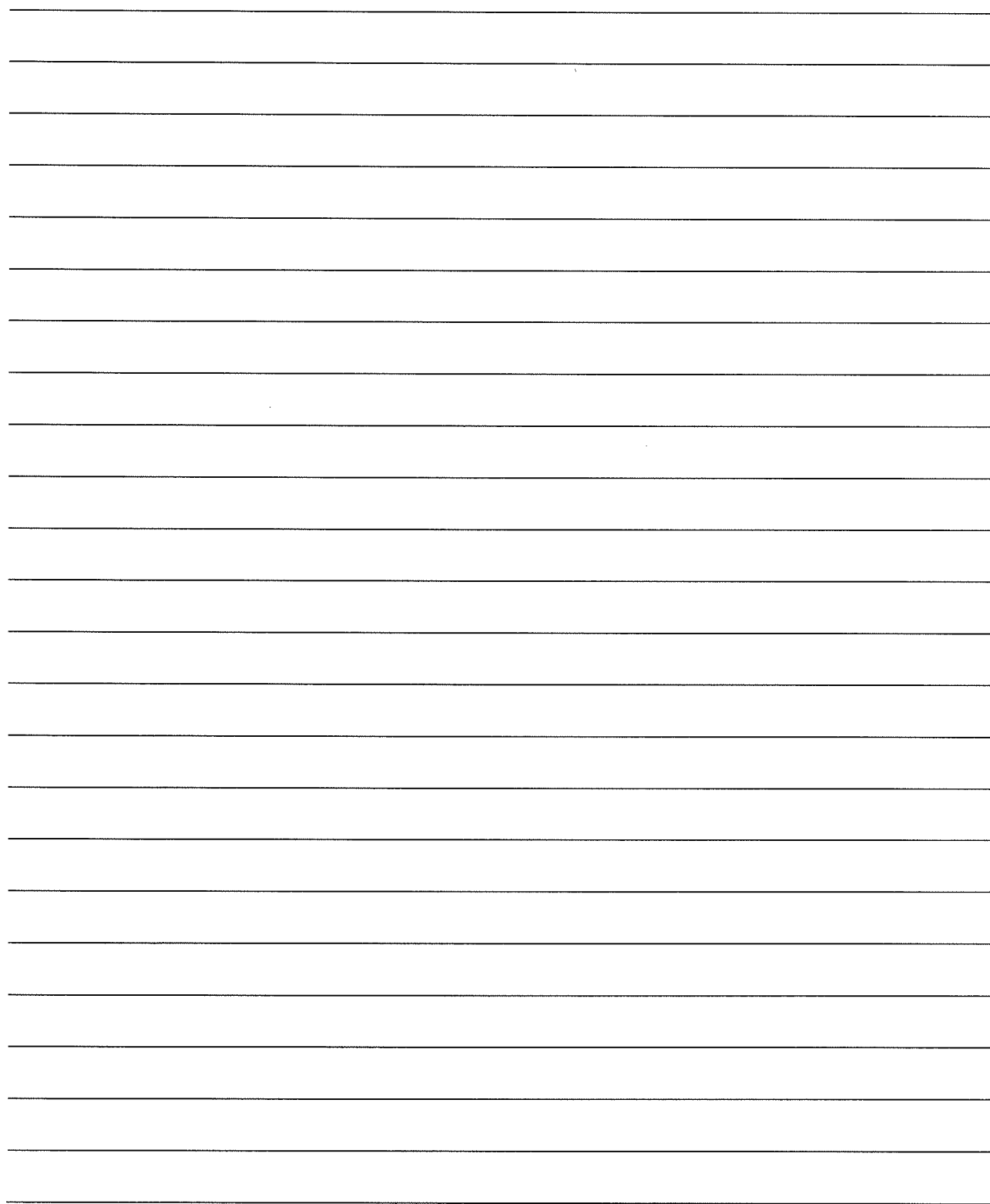
Draw a class diagram for your design on the next page.

5. [2pts] What effect did your new design have on the coupling between class IntelligentPilot and class CessnaControls.
- a. Reduced their coupling
 - b. Increased their coupling
 - c. Had no effect on their coupling

Write your answer to question 4 here.







8. [12pts] Consider the following web server, which is similar to a Java EE Web Container. Note that it is *serial* in that it is not multithreaded.

```
public class SerialServer {
    public static void main(String[] args) throws ... {
        while (!isHalt()) {
            // Receive an HTTP request from the network
            HttpServletRequest request = acceptRequest();
            // Look up the appropriate servlet, using the URL
            // contained in the request
            HttpServlet servlet = retrieveServlet(request);
            // Create an instance of HttpServletResponse
            HttpServletResponse response = createResponse();
            // Have the servlet service the request
            servlet.service(request, response);
        }
        ...
    }
}
```

Your job is to re-implement this class as a multithreaded web server. Your server should be more responsive than the serial version because it services requests in parallel. (Note that only the main thread should accept requests.)

Here is an example of how to spawn a thread in Java:

```
// To define a thread's behavior, you must define a class that
// implements the Runnable interface
public class MyRunnable implements Runnable {
    // Because it implements Runnable, it must define a run() method
    public void run() {
        // This is where each thread's behavior goes
        ...
    }
}

public class MyMultithreadedApp {
    public static void main(String[] args) {
        // Create a thread that will have the behavior from MyRunnable
        Thread myThread = new Thread(new MyRunnable());
        // Spawn the thread and have it execute MyRunnable.run()
        myThread.start();
        // The main thread continues on its merry way, while the
        // other thread runs concurrently
        ...
    }
}
```

Write the code to implement your server on the next page (or two). Hint: You should have a **Multi-threadedServer** class with a **main()** method, and a class **MyRunnable** that contains the behavior that each thread should exhibit. Hint: You may have to use instance variables of **MyRunnable** to pass objects from the **main()** method to the **run()** method.

Write your answer to question 8 here.

```
public class MyRunnable implements Runnable {  
    public void run() {  
        servlet.service(request, response);  
    }  
  
    private HttpServlet servlet;  
    private HttpServletRequest request;  
    private HttpServletResponse response;  
    public MyRunnable(HttpServlet s,  
                      HttpServletRequest rq,  
                      HttpServletResponse rp) {  
        servlet = s;  
        request = rq;  
        response = rp;  
    }  
}
```

~

```
public class MultithreadedServer {  
    public static void main(String[] args) {  
        while (!isHalt()) {  
            HttpServletRequest request = acceptRequest();  
            HttpServlet servlet = retrieveServlet(request);  
            HttpServletResponse response = createResponse();  
            Thread myThread = new Thread(new MyRunnable(  
                servlet, request, response));  
        }  
    }  
}
```

(more next page)

Continue your answer to question 8 here.

```
myThread.start();
```

```
    }
```

```
    }
```

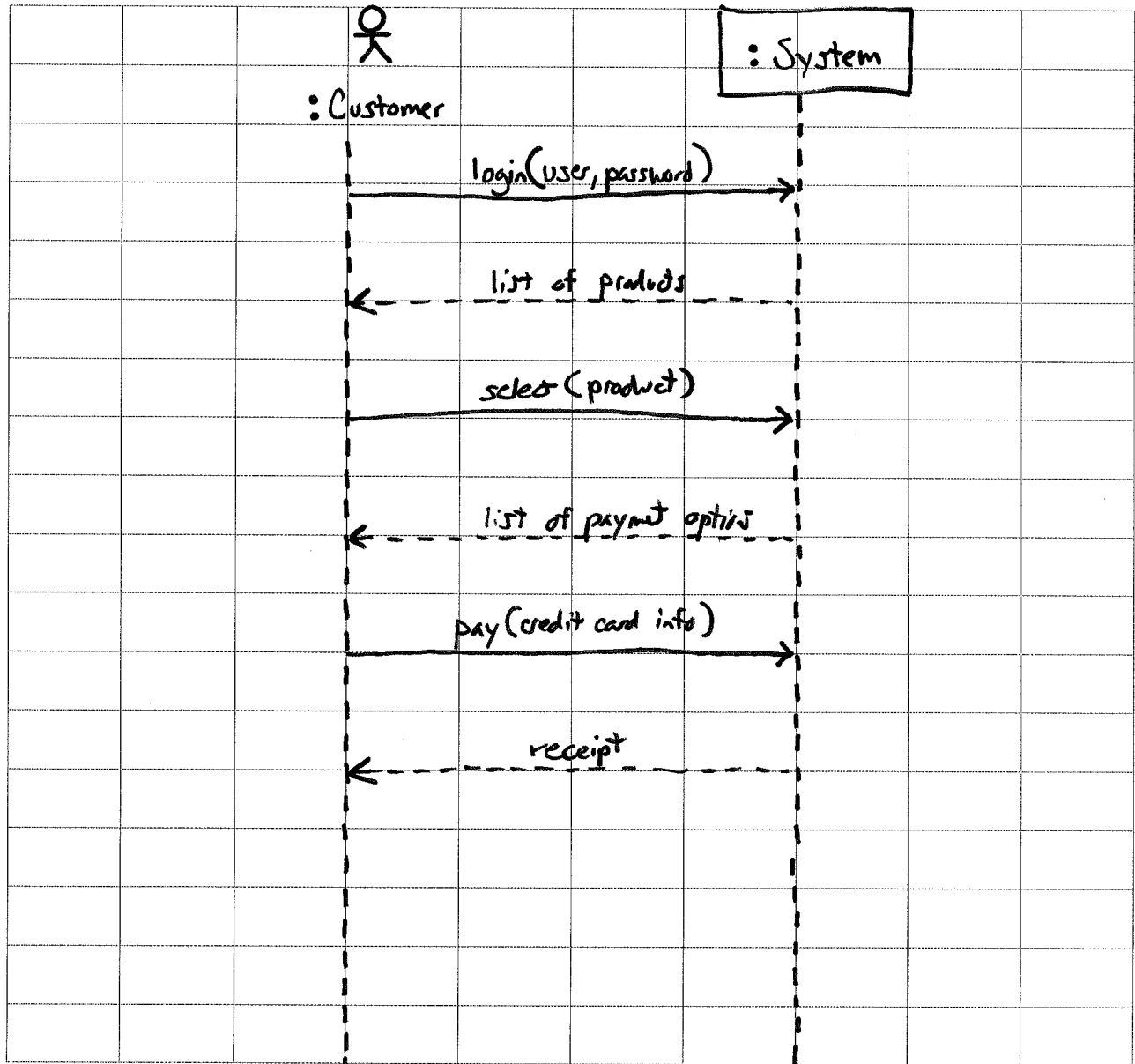
```
    }
```

```
}
```

9. [10pts] Given the following scenario, draw a system sequence diagram.

1. The customer enters his/her login and password.
2. The system validates the login information.
3. The system displays a list of products.
4. The customer selects a product.
5. The system displays a list of payment options.
6. The customer enters his/her credit card info.
7. The system charges the purchase to the credit card.
8. The system displays a purchase receipt.

(Write your answer here.)



10. [2pts] SRP is short for:

- a. Software Requirements Process
- b. Sequential Response Protocol
- c. Server Receive Packet
- d. Single Responsibility Principle
- e. None of the above

Extra Credit Questions

11. [1pt] May I use your project code as an example in future classes? Having such examples is extremely helpful to both me and students.

- a. Yes, and be sure to credit me.
- b. Yes, but I do not want my name attached to it.
- c. No

12. [1pt] Tell me one thing that you particularly liked about this course.

13. [1pt] Tell me one thing that you would change about the course.