Student ID_____Student Name_____

Advanced Software Development DE Final Exam July 6, 2013

PRIVATE AND CONFIDENTIAL

1. **Allotted exam duration is 2 hours.**
2. **Closed book/notes.**
3. **No personal items including electronic devices (cell phones, computers, calculators, PDAs).**
4. **Cell phones must be turned in to your proctor before beginning exam.**
5. **No additional papers are allowed. Sufficient blank paper is included in the exam packet.**
6. **Exams are copyrighted and may not be copied or transferred.**
7. **Restroom and other personal breaks are not permitted.**
8. **Total exam including questions and scratch paper must be returned to the proctor.**

5 blank pages are provided for writing the solutions and/or scratch paper. All 5 pages must be handed in with the exam

BE VERY CAREFUL WITH THE GIVEN 2 HOURS AND USE YOUR TIME WISELY. THE ALLOTED TIME IS GIVEN FOR EVERY QUESTION.

Write your name and student id at the top of this page.

Question 1 [40 points] {55 minutes}

Suppose you need to make a relatively simple CRUD (Create, Read, Update, Delete) application that manages employees. The application has the following requirements:

- You can add new employees, update existing employees, remove employees, view an employee, search employees and get a list of all employees.
- You should be able to undo/redo the add, update and delete action
- All employees are stored in the database
- You want to log all add, update and delete actions in a logfile

Just when you want to start with the design of this application you get a request from another customer to write a CRUD ProductService application that manages products.

This application has the following requirements:

- You can add, update, remove and view a product.
- You can search products.
- You should be able to undo/redo the add, update and delete action
- All products are stored in an XML file
- You need to send an email to the sales manager whenever a product is added or deleted.
- You want to log all CRUD actions in a logfile
- You should be able to print an overview of all available products.

Because you need to implement 2 similar CRUD applications you decide to design a general CRUDService framework so that you can reuse this framework for both CRUD applications. This CRUDService framework should be very flexible so that we can reuse it for other CRUD-like applications with similar requirements.

Draw in one class diagram the ProductService application using the general CRUDService framework. In the class diagram, show clearly which classes are within the framework, and which classes are outside the framework. Make sure that the framework contains all the general implementation so that the classes outside the framework are as simple as possible.

Make sure you add all necessary UML elements (attributes, methods, multiplicity, etc) to communicate the important parts of your design.

Question 2 [40 points] {45 minutes}

Suppose we need to design a library system framework which allows us to manage Books, borrowings and reservations. Consider the following requirements for this framework:

- The framework should record customer information: name, phone, email, street, city, zip.
- The framework should record book information: title, isbn, author.
- When a customer checks out We should be able to record (checkout-date, return-date, fee) and handle the different books that a customer borrows.
- We should be able to record and handle the reservations of a customer.
- It should be easy to plugin different algorithms to compute the fee for books that are returned to late.
- We can have multiple copies of the same book title. Every copy has a unique scancode.
- The framework should support the ability to create different book categories and subcategories. For example, we can have a category Computer book with subcategory UML books

Draw the class diagram of this library framework. In the same class diagram, show how this library framework is used by a library application with the following requirements:

- The library application should support both Books and DVD's.
- The library application should support 2 different ways to compute the fee for books that are returned too late:
- During the summer break we calculate the fee in a different way as during the regular days.
- The library application should support foreign customers. The application should record the country for these foreign customers.

Draw in one class diagram the library application using the general library framework. In the class diagram, show clearly which classes are within the framework, and which classes are outside the framework.

Make sure you add all necessary UML elements (attributes, methods, multiplicity, etc) to communicate the important parts of your design.

Question 3 [15 points] {10 minutes}

Consider the following application that uses a dynamic proxy:

```
public interface IVehicle {
       void start();
}


public class Car implements IVehicle {
      private String name ="Herbie";

      public void start() {
            System.out.println("Car " + name + " started");
      }

}


public class Logger implements InvocationHandler {
      private Object v;

      public Logger(Object v) {
            this.v = v;
      }

      public Object invoke(Object proxy, Method m, Object[] args)
                  throws Throwable {
            System.out.println("Logger: " + m.getName());
            Object object= m.invoke(v, args);
            return object;
      }
}


public class Notifier implements InvocationHandler {
      private Object v;

      public Notifier(Object v) {
            this.v = v;
      }

      public Object invoke(Object proxy, Method m, Object[] args)
                  throws Throwable {
            System.out.println("Notifier: " + m.getName());
            Object object= m.invoke(v, args);
            System.out.println("Notifier: " + m.getName());
            return object;
      }
}
```

What is the output written to the console when we run the following application?

```
public class Application {
	public static void main(String[] args) {
		IVehicle c = new Car();
		ClassLoader cl = IVehicle.class.getClassLoader();
		IVehicle v1 = (IVehicle) Proxy.newProxyInstance(cl, new Class[]
		    { IVehicle.class }, new Logger(c));
		IVehicle v2 = (IVehicle) Proxy.newProxyInstance(cl, new Class[]
		    { IVehicle.class }, new Notifier(v1));
		v2.start();
	}
}
```

Your answer:

Question 4 [5 points] {10 minutes}

Describe how the factory pattern relates to one or more of the SCI principles you know. Your answer should be about half a page, but should not exceed one page (handwritten). The number of points you get for this questions depend on how well you explain the relationship between the factory pattern and the principles of SCI.

Your answer: