Student ID__________________Student Name____________________________________

Advanced Software Development DE Final Exam August 13, 2011

PRIVATE AND CONFIDENTIAL

1. **Allotted exam duration is 2 hours.**
2. **Closed book/notes or open 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.**

7 blank pages are provided for writing the solutions and/or scratch paper. All 7 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 [25 points] {30 minutes}

We need to design a contact framework which we can use to write contact applications that allow us to manage our contacts. The framework has the following requirements:

- We should be able to add new contact, delete contacts and search contacts.
- For every contact we need to store its name, phone, email, fax number, company name, company phone, street, city and zip code.
- The framework should also support the functionality to show the list of the last 10 executed actions ("add contact" actions and "delete contact" actions), and we should be able to select one of these actions and undo the selected action. Search actions are not shown in this list.

Now we want to use this framework to implement a contact application. This contact application has the following additional requirements:

- We should also be able to edit existing contacts
- We should also be able to undo edit actions
- The contact application should support international addresses for contacts, so for every contact, we should also store the country (besides the street, city and zip).

Draw in **one class diagram** the design of the contact application using the framework.
So this class diagram should show the design of the framework, and the design of the contact application. 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 2 [40 points] {50 minutes}

First we have to design a car rental application for one of our customers with the following requirements:

- The application keeps track of which customers (name, address) rents which car (licenceNumber, brand, type, price_per_day) on which days.
- The application keeps track of the list of credit cards that a customer may have
- The application keeps track of payments(amount, date)
- Car rentals can only be paid with credit card.
- Cars are categorized in categories (economy, business(standard, full size, specialty), minivan, suv, etc)

a. Draw the **class diagram** the design of the car rental application

Then another customer wants us to design a vacation house rental application with the following requirements:

- The application keeps track of which customers (name, address) rents which vacation house (address, price_per_week) on which days.
- The application keeps track of payments(amount, date)
- House rentals can be paid with credit card, cash or with a check. If you pay by check, the application should store the name on the check, the check number and the bank name on the check.
- Vacation houses are categorized in categories (condo (1 room, 2 rooms, etc), house(on the beach, 1 room, 2 rooms, etc), villa, etc)

b. Draw the **class diagram** the design of the vacation house rental application

Now another customer wants us to design a tool rental application with the following requirements:

- The application keeps track of which customers (name, address) rents which tools (toolnumber, toolname, price_per_day, price_per_week, price_per_month) on which days.
- A customer can rent out multiple tools in one rental.
- The application keeps track of payments(amount, date)
- Rentals can be paid with credit card, cash, check or with bank wire. If you pay by check, the application should store the name on the check, the check number and the bank name on the check. If you pay with bank wire, the application should store the name of the bank.
- Tools are categorized in categories (drilling tools, sawing tools, etc)
- Tool rentals can be insured but insurance is not required. If a tool rental is insured, then the application should keep track of the maximum amount that is insured and the price that has to be paid for the insurance.

Because this is our third rental application, we decide to make a rental framework.

The framework should support the following requirements:

- The framework keeps track of which customers (name, address) rents which rental-products on which days.
- A customer can rent out multiple rental-products in one rental.
- The application keeps track of payments (amount, date)
- Rentals can be paid with credit card, cash, check or bank wire. If you pay by check, the application should store the name on the check, the check number and the bank name on the check. If you pay with bank wire, the application should store the name of the bank.
- Rental-products are categorized in categories
- The framework supports all possible ways to specify the rental price: price per hour, price per half day, price per day, price per week, price per month, price per year, price per 2 weeks, etc.

c. Draw the **class diagram** the design of the tool rental application using the framework.

So this class diagram should show the design of the framework, and the design of the tool rental application. In the class diagram, show clearly which classes are within the framework, and which classes are outside the framework.

Question 3 [30 points] {30 minutes}

We want to design a points award framework that allows us to write applications that records the number of points in a points award program like frequent flyer miles programs or hotel reward programs. The framework should support the following requirements:

- Members should have a certain status, like bronze, silver and gold, and depending on the status you get a certain number of bonus points
- The possible statuses are application specific
- Every status knows its next status
- Customers can subscribe themselves to notifications on certain events, for example when their account state changes (from silver to gold for example) or when new points are added to their account. Customers can decide themselves how they want to be notified, for example by email or by SMS. The framework should support both notifications by email and notifications by SMS.
- It should be easy to add more notification options, for example by regular mail.
- The history of added points should be available. We should be able to see how many points are added at what date.

Draw the **class diagram** a certain points award application using the points award framework. The points award application should support the following requirements:

- The award program supports 2 types of accounts: Starter accounts and Premium accounts. Premium accounts receive 2 times more points as Starter accounts.
- If new points are added, customers can be notified by email, SMS or receive a letter by regular mail.
- The points awards program allows us to see the history overview that shows the date, the number of points that are added and a description that describes why we get these points.

In the class diagram, show clearly which classes are within the framework, and which classes are outside the framework.

Question 4 [5 points] {10 minutes}

Describe how the Factory method pattern relates to one of the SCI principle you know. Your answer should be about half a page, but should not exceed one page (handwritten).
Write clearly – if I cannot read it I cannot give you any points for it.