Student ID_____Student Name_____

Advanced Software Development DE Final Exam

June 27, 2015

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.**

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 [35 points] {40 minutes}

During this DE course we used Sakai as a Learning Management System (LMS).
Suppose you have to design a framework for a LMS with the following requirements:

Requirements for the LMS framework:

- The LMS framework supports multiple courses.
- A course can be taken by multiple students
- The professor should be able to add assignments to a course. An assignment has a title, openDate, dueDate, instructions and can contain multiple attachments.
- Students can see the assignments, and submit their solution for every assignment. The system keeps track of the submission date. Students can submit multiple attachments to their solution.
- The professor can grade the solutions from the students. The LMS framework should support different grading options like letter grade, point grade or pass/fail.
- When the professor sets the grade for a solution, then the student will receive an email about this. An attachment is stored on the local file system.

Requirements for the application using the LMS framework:

- The application using this LMS framework (called MUM LMS) wants the possibility that an assignment contains multiple sub-assignments. A sub-assignment has the same openDate and dueDate as its parent assignment, but has its own title and instruction. A sub-assignment can also have multiple attachments.
- When the professor sets the grade for a solution, then the student will receive either an email, a WhatsApp message or a SMS messages about this, depending on the preferences of the student.
- It should be easy for LMS applications using this framework to add their own grading option, for example a checkmark.

Draw the class diagram of your design. In the class diagram, show clearly which classes are within the framework, and which classes are outside the framework (based on the requirements for the framework, and the framework best practices we studied in this course) . **Make sure you add all necessary UML elements (interfaces, abstract classes, attributes, methods, multiplicity, etc) to communicate the important parts of your design. You only need to show the domain model of the point awards framework and application, you do not need to worry about GUI classes, service classes, database classes, etc.**

Question 2 [20 points] {20 minutes}

Suppose we have a client server application in which the client sends mathematical requests to the server. The server computes the mathematical request, and returns the result of the request. In our current application, the client can send only 4 types of simple requests:

1. The add operation of 2 numbers (the client sends: add, 3, 6 and the server responds with 9)
2. The add operation of 3 numbers (the client sends: add, 3, 6, 8 and the server responds with 17)
3. The multiply operation of 2 numbers (the client sends: multiply, 3, 6 and the server responds with 18)
4. The multiply operation of 3 numbers (the client sends: add, 3, 6, 8 and the server responds with 144)

The current implementation of the server is as follows:

```
public class ActionReceiver {

	public Response receiveAction(Request request){
		Response response = new Response();
		int result=0;

		if (request.getOperation().equals("add")){
			if (request.getParams.size() == 2){
				result= request.getParam(1) + request.getParam(2);
			}
			else{
				if (request.getParams.size() == 3){
					result= request.getParam(1) + request.getParam(2)
						 + request.getParam(3);
				}
			}
		}
		else{
			if (request.getOperation().equals("multiply")){
				if (request.getParams.size() == 2){
					result= request.getParam(1) * request.getParam(2);
				}
				else{
					if (request.getParams.size() == 3){
					      result= request.getParam(1) *
					        request.getParam(2) * request.getParam(3);
					}
				}
			}

		}
		response.setResult(result);
		return response;
	}
}
```

You get the task to modify the application so that the client can send any mathematical request to the server and the server will return the result. A Mathematical request always consist of a mathematical operation and one or more numbers. Here are some examples:

Request	Response
Subtract, 22, 11	11
Sqrt, 5	25 (square root)
Add, 3, 4, 5, 6, 7	25

At the moment the application should support 388 different mathematical operations, but it should be very easy to add new mathematical operations to this applications.

Draw the class diagram of your design of the server application (NOT THE CLIENT). Make sure you add all necessary UML elements (interfaces, abstract classes, attributes, methods, multiplicity, etc) to communicate the important parts of your design.

Question 3 [40 points] {50 minutes}

Suppose we need to design a bankaccount framework with the following requirements:

We should be able to create a new account for a customer of the bank.
On an existing account, we can deposit and withdraw money.
We can also transfer money to another account.
It should be possible to see the history of transactions (deposits, transfers, withdraws) for every account.
We also need undo/redo functionality for deposit and withdraw actions
The framework should supports Savings and Checking accounts, but it should be easy to add other accounts.
For all accounts we should be able to add interest, and the interest algorithm depends on the account type (checking, saving, …). The framework should support a low interest algorithm for checking accounts, and a high interest algorithm for saving accounts. It should be easy to add other interest algorithms.
Whenever the Account balance is changed we want the following actions:

- A Logger class should log every change to an Account
- A SMSSender should send a SMS at every change to an Account.

It should be easy to add more actions whenever the account changes.
All account data should be saved in the database.

a. Draw the class diagram of your bankaccount framework.
b. Draw the sequence diagram of the scenario where a customer deposits $100 to accountnr 342

Make sure you add all necessary UML elements (interfaces, abstract classes, attributes, methods, multiplicity, etc) to communicate the important parts of your design.
Make sure that your design follows the design principles we studied in this course.

Question 4 [5 points] {10 minutes}

Describe how AOP 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 AOP and the principles of SCI.