Student ID_____Student Name_____

Advanced Software Development DE Final Exam

May 26, 2018

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.**

6 blank pages are provided for writing the solutions and/or scratch paper. All 6 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 [20 points] {20 minutes}

Suppose you need to develop an application that is used in a self-driving car. Because this software supports many different car brands and car types, you need to design a flexible framework for this.
You have the following design problem:
A self-driving car has many sensors. The signals of these sensors should be handled by certain methods in your code. For example the camera in the middle front of the car generates a new image every 2 seconds and this image should be handled by the method processMiddleFrontCameraImage(). The images from the left front camera should be handled by the method processLeftFrontCameraImage(). The GPS device sends new gps coordinates every second which should be handled by the method processGPSSignal()
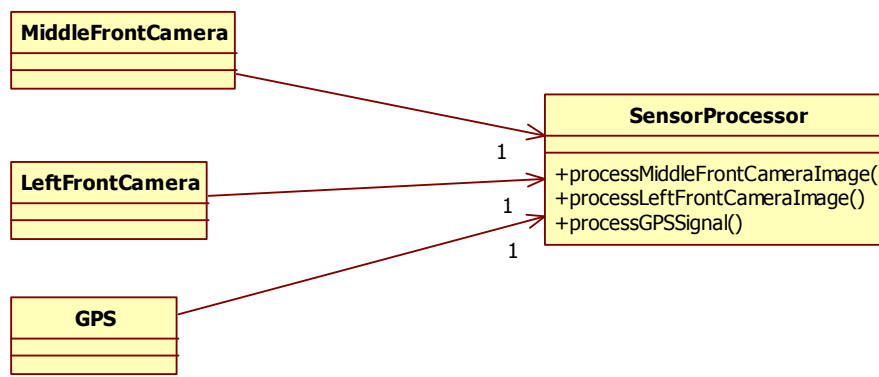
Your first design could be something like the diagram above, but the problem with this design is that we need to change the SensorProcessor class every time we add a new sensor. And because every car type has different sensors you have to come up with a better design.

Draw the class diagram of your design **that allows us to support different sensors, and it should be easy to add more sensors without having to make a lot of modifications in the code**. Make sure that your class diagram only shows the design that solves the given problem

Question 2 [10 points] {10 minutes}

Explain clearly the difference between the Factory Method and the Abstract Factory pattern.

Question 3 [10 points] {10 minutes}

Suppose a ProductService class needs to call methods on the ProductDAO class.

We could implement the ProductService as follows:

```java
public class ProductService {

  private ProductDAO productDAO = new ProductDAO();

  public void AddProduct(String productNumber, String name, double price){
        productDAO.save(new Product(productNumber, name, price));
  }

  //other methods are not shown

}
```
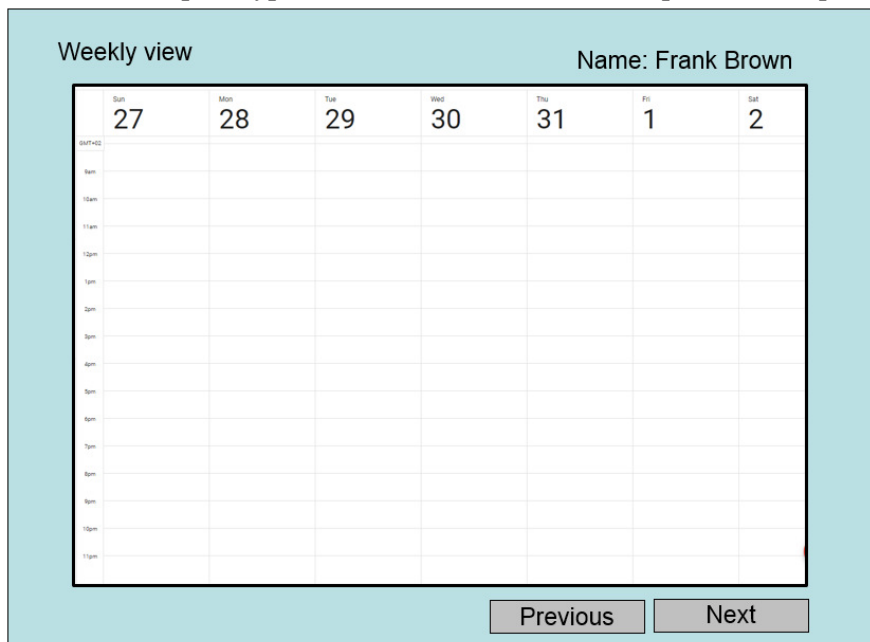
    a. What is the problem with the given code?

    b. Show how can we improve this code?

Question 4 [55 points] {70 minutes}

Suppose we need to design a calendar framework that allows us to quickly create all kind of calendar applications. The framework has the following requirements:

- Users of the application can create their own calendar.
- They can add and remove appointments to and from their calendar.
- Every appointment contains the following data: startDate, startTime, endDate, endTime, name of the appointment, and description of the appointment.
- Users first have to login with their username and password to access their calendar.
- The calendar can be shown with multiple views (monthly, weekly, daily)
- It should be easy to add more views (yearly, …)
- Users can choose to show multiple views at the same time. So a user can choose to show both the monthly and the weekly view at the same time. This means when a user adds a new appointment, this appointment should show up both on the monthly view as well on the weekly view
- Users can be notified 1 hour before the appointment is scheduled.
- Users can choose how to be notified: by email or by whatsapp message.
- It should be easy to add more ways of notification (SMS,…)
- All calendar data should be stored in a database.
- The framework also supports undo/redo for both creating and removing an appointment

Here are some prototype screens that have been developed in the requirements phase:

Monthly view
Name: Frank Brown

Sun 27	Mon 28	Tue 29	Wed 30	Thu 31	Fri Jun 1	Sat 2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Previous Next

Create new appointment

Start date
End date
Start time
End time
Name
Description

Create Cancel

a. Draw the class diagram of your calendar framework.
b. Draw the sequence diagram of the following scenario
   1. The user first creates a new appointment, and this new appointment is shown on both the monthly view and the weekly view
   2. Then the user selects the undo action, so the appointment disappears from both the monthly view and the weekly view

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 5 [5 points] {10 minutes}

Describe how the principles of a framework relate 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 question depends on how well you explain the relationship between the principles of a framework and the principles of SCI.

Your answer: