Name:_____**Registration-Nb.:**_____

Task 1 Software Development Processes

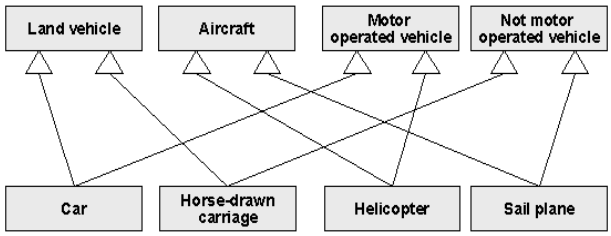
1.1 List three reasons why to use a Software Development Process. [/ 3]

1.2 Object Oriented Software Development [/ 15]

a. What is the goal of Object Oriented Analysis? [/ 2]

b. What are the main differences between an Object Oriented Analysis Model and an Object Oriented Design Model? [/ 2]

c. Following Object Oriented Analysis Model is given. Explain one possible reason or consideration, why the Object Oriented Design Model should look different. [/ 1]



d. Create an Object Oriented Analysis Model based on following description [/ 5]

> A weather monitoring station contains a temperature sensor, an humidity sensor, and a pressure sensor. Additional sensors could be added in future.
> The weather monitoring station processes the observed data from the sensors and displays the results on a display device.

e. What is a component in Object Oriented Design? [/ 2]

f. List and explain three Object Oriented Design Principles [/ 3]

1.3 Agile Software Development [/ 6]

a. Describe 3 practices of Extreme Programming. [/ 3]

b. List the main differences between the Rational Unified Process and Agile Software Development Processes (Extreme Programming) [/ 3]

Task 2 Requirements Analysis

2.1 Functional or non functional (required constraints) requirements? [/ 3]

Requirement	Functional	Non functional
Mamegoma Mark 2 robot should clean sofa, bed, and bin		
The new cipher machine model should encrypt 1500 messages per hour		
The Protoss Zealot is only able to attack unit on the ground		
The command “get ready” should be given to 1,000 soldiers in 5 minutes		
The command “next” will change the state of the canvas object to the next color.		
We need to create a house in many styles with the steps - Create building framework - Create floor - Create wall - Create roof		

2.2 Use Case diagram [/ 3]

Create a Use Case diagram based on following description:

- A customer uses an elevator to move up or down
- If the customer is moving up or down, she optional could activate an alarm
- If the alarm is activated, the elevator sounds an alarm signal

Task 3 UML diagrams

3.1 What does the abbreviation UML mean? [/ 1]

3.2 Class diagrams [/ 3]

Please create class diagrams for following descriptions:

The abstract class `F` has a private attribute `size` and a public method `getsize()`	The classes `A` and `B` inherit from the abstract class `R`. `A` contains `R` as aggregate, so `A` has any `R`.	`C` is related to any number of `D`

3.3 Sequence diagram [/ 4]

Please create a sequence diagram representing the following facts:

> A customer `goodCustomer` sends to the company `niceCompany` an order containing the attribute `text`. The company `niceCompany` sends back a detailing request to the customer `goodCustomer`. The customer `goodCustomer` resends to the company `niceCompany` an order containing the attribute `detailedText`.

Task 4 Design Pattern – General

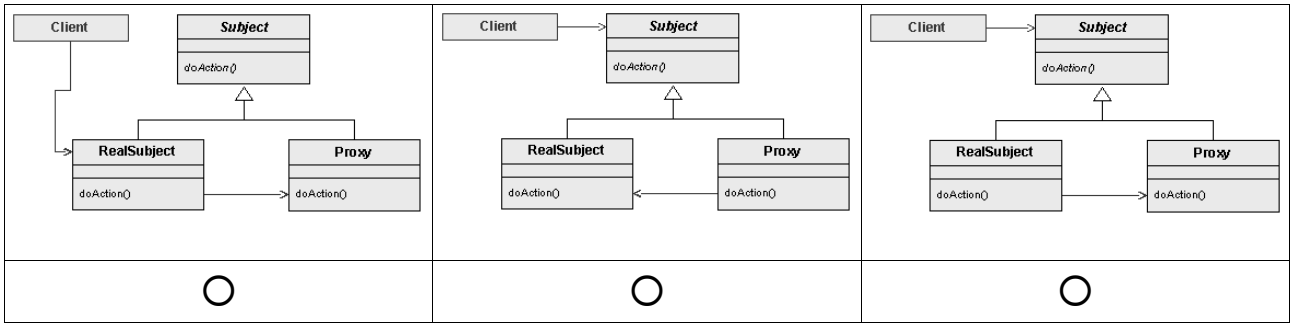
4.1 Characteristics of Design Pattern [/ 1]

Which of the following statements describe the characteristics of Design Pattern (more than one answer could be selected)?

1. Design Pattern offer a vocabulary for talking about design. ○
2. Design Pattern identify key aspects of a common design structure. ○
3. A Design Pattern is an implementation specific finished design ○
4. Design Pattern avoid reuse of design and code ○

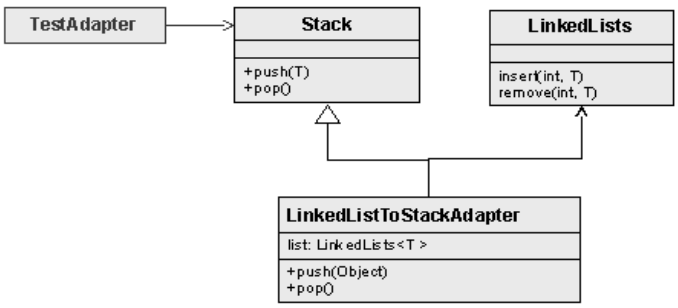
4.2 Structures of Design Pattern [/ 7]

a. Which diagram shows the Proxy Pattern? [/ 1]

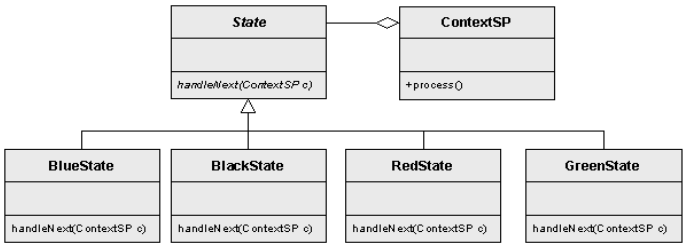


○ ○ ○

b. Which special kind of which Design Pattern is described here? [/ 1]



c. Which Design Pattern is used here? [/ 1]



d. Design a UML diagram showing the structure of the Observer Pattern [/ 4]

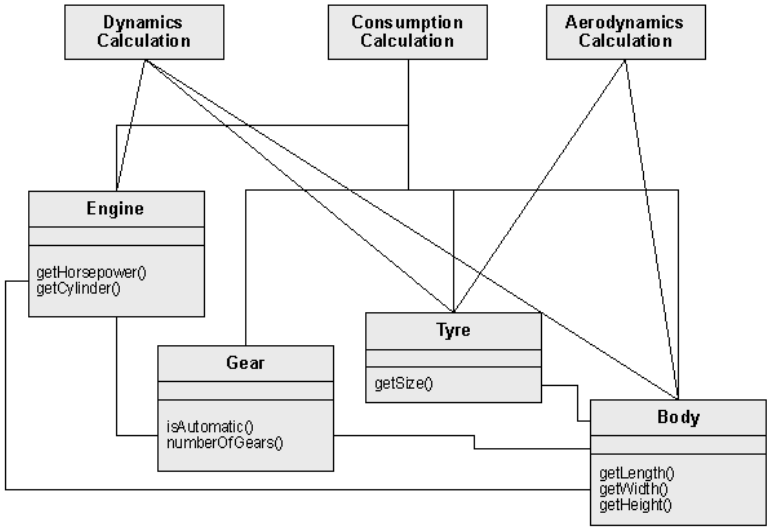
4.3 Categories of Design Pattern [/ 4]

a. List 4 Structural Design Pattern [/ 2]

b. Describe the intent of 1 Behavorial Design Pattern [/ 2]

Task 5 Façade Design Pattern

Following UML class diagram is given:



5.1 Design discussion [/ 2]

a. Why should you use the Façade Design Pattern here? [/ 1]

b. If you use the Façade Design Pattern, what will be improved? [/ 1]

5.2 Create a new UML class diagram using the Façade Design Pattern [/ 4]

5.3 Related Pattern [/ 2]

What is the difference between the Adapter pattern and the Façade pattern?

Task 6 Singleton Design Pattern

6.1 Describe the intent of the Singleton Design Pattern [/ 1]

6.2 Structure of the Singleton Pattern [/ 2]

Design and describe the structure of the Singleton Pattern in a UML class diagram

6.3 Which Design Pattern could be used with the Singleton Pattern? [/ 2]

6.4 Give an example, when the use of the Singleton Pattern is applicable [/ 1]

6.5 Singleton implementation [/ 2]

Attached you see a proposal for an implementation

```
public class Singleton {
   private final static Singleton instance = new Singleton();

   private Singleton() {}

   public static Singleton getInstance() {
     return instance;
   }
}
```

In another part of the code you see following line:

```
mySingleton = new Singleton();
```

a. What is your comment to this line? [/ 1]

b. How would you update this line? [/ 1]

Task 7 Abstract Factory Design Pattern

7.1 Describe the intent of the Abstract Factory Pattern [/ 2]

7.2 Implementation example [/ 20]

Following code sample is given:

```
public abstract class CarFactory {
    public static CarFactory getFactory(String model) {
        if (model == "BMVV") {
            return new BMVVFactory();
        } else if (model == "Morecedes") {
            return new MorecedesFactory();
        }



        else
            return null;
    }

    public abstract LuxuryClassCar createLuxuryClassCar();



}

public class MorecedesFactory extends CarFactory {
    public LuxuryClassCar createLuxuryClassCar() {
        return new SpecialClass();
    }



}

public class BMVVFactory extends CarFactory {
    public LuxuryClassCar createLuxuryClassCar() {
        return new SevenSeries();
    }



}
```

```
public abstract class Car {
    public abstract void produce();
}

public abstract class LuxuryClassCar extends Car {
}

public class SevenSeries extends LuxuryClassCar {
    public void produce() {
        System.out.println("Producing a nice BMVV Seven Series Luxury Car!");
    }
}

public class SpecialClass extends LuxuryClassCar {
    public void produce() {
     System.out.println("Producing a nice Morecedes Special Class Luxury Car!");
    }
}

public class Client {
    public static void main(String[] args) {
        CarFactory factory = CarFactory.getFactory("BMVV");
        Car car = factory.createLuxuryClassCar();
        car.produce();
    }
}
```

a. Explain, which method in the code is a Factory Method. [/ 3]

b. What will be the result of this code, if you compile it? [/ 2]

c. What do you have to change in the code if you would like to get the output `"Producing a nice Morecedes Special Class Luxury Car!"`? [/ 2]

d. Design for the given code sample the corresponding UML class diagram [/ 4]

e. Which class is taking which role of the Abstract Factory Design Pattern? [/ 3]

Abstract Factory Pattern	Code sample
Abstract Factory	
ConcreteFactory1	
ConcreteFactory2	
AbstractProductA	
ProductA1	
ProductA2	

f. Add another CarFactory `"OOOOFactory"`, which could produce a luxury class car named `"OOOOEight"`.
Write the corresponding code to be added into the code sample on pages 12 and 13, add the necessary code for new classes here: [/ 6]

7.3 Consequences of the Abstract Factory Design Pattern [/ 2]

Describe the consequences of the Abstract Factory Design Pattern.
What are the benefits and what are the disadvantages?