Software Design

Final exam

January 27th, 2017.

Student ID	*Name and Surname*	*Handwritten signature*

1. **(1 point)** List all possible ways to express system requirements.

2. **(1 point)** List generic activities in requirements engineering.

3. **(1 point)** Name at least two properties of agile methodology for software development.

4. **(1 point)** Name the key property of waterfall model of software development.

5. **(1 point)** Specify how the user requirements have been expressed in your project assignment documentation.

6. **(2 points)** The following source code describes the class Shape2D and two classes Square and Rectangle which inherit the class Shape2D:

```
/**
 * Class Shape2D
 */
public abstract class Shape2D {

    public abstract int Surface(int params[]);

    public void Identify()
    {
       //method for console printing
        Print("I am Shape 2D");
    }
}
```

```
/**
 * Class Square inherits class Shape2D
 */
public class Square extends Shape2D{

    public int Surface(int params[]){
        return params[0]*params[0];
    }

    public void Identify(){
        Print("I am Square");
    }
}
```

```
/**
 * Class Rectangle inherits class Shape2D
 */
public class Rectangle extends Shape2D{

    public int Surface (int params[]){
        return params[0]*params[1];
    }
}
```

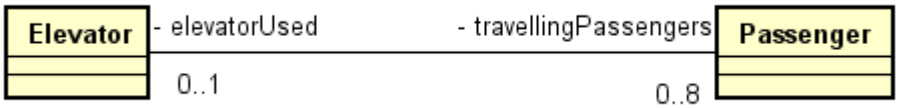
a) (1 point) Explain if the following code line in the `main()` method is correct or not:

```
Square s = new Shape2D();
```

b) (1 point) What will be written in the console when the `main()` method is executed. The source code of the method is given below.

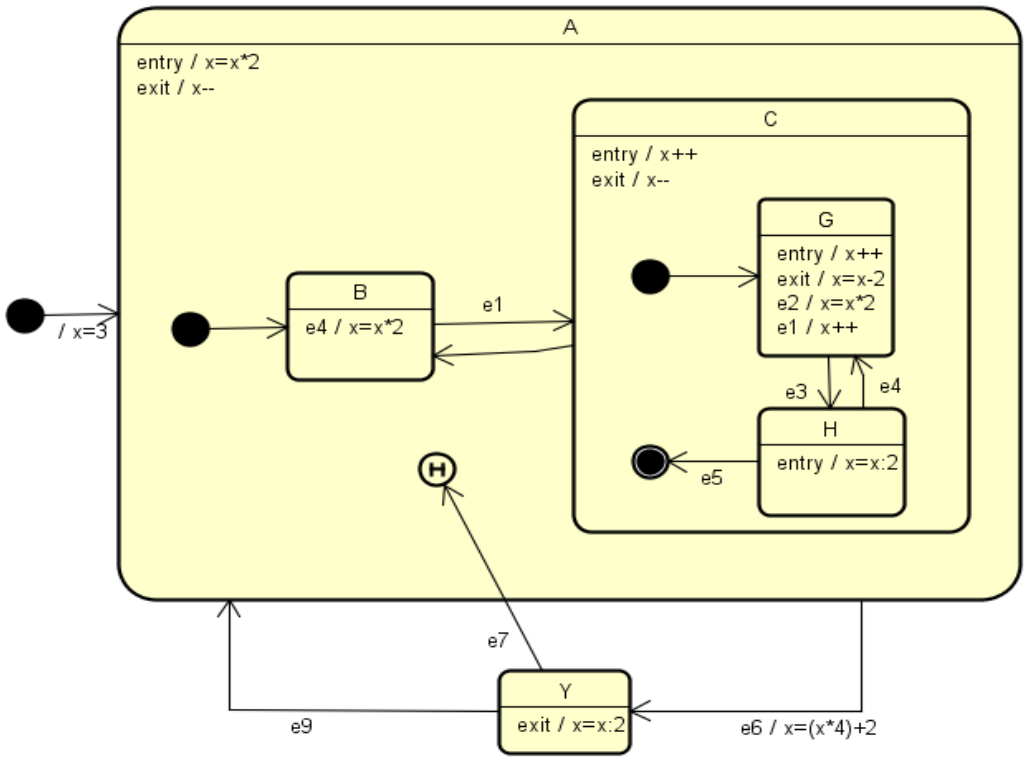
```
public static void main(String[] args) {

        Shape2D s = new Square();
        Shape2D r = new Rectangle();

        int []params_s = new int[1];
        int []params_r = new int[2];

        params_s[0]=5;
        params_r[0]=2;
        params_r[1]=3;

       Print(s.Surface(params_k));
        Print(r.Surface(params_p));

        s.Identify();
        r.Identify();
    }
```

7. **(1 point)** According to the UML diagram below answer the following:
    a. What is the maximum number of passengers which can travel in an elevator?
    b. Must the passenger travel in an elevator at all?



8. **(1 point)** List all fundamental elements of UML Sequence diagram and mark them on a hypothetical example.

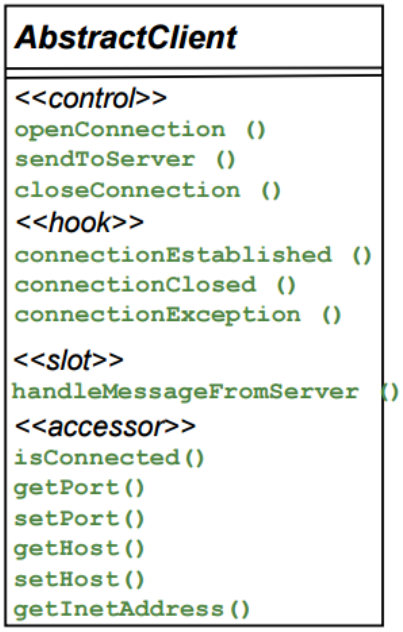
9. **(2 points)** For the given UML Statemachine diagram determine the value of variable *x* upon the execution of events: **e1, e3, e6, e7** (*x* = 3 at the beginning).



10. **(1 point)** Describe the organization of application program in a service-based architecture.

11. **(1 point)** List and describe in short minimal elements used to describe design patterns.

12. **(1 point)** To which of the following groups of methods in the AbstractClient class of the OCSF architecture belong the methods which <u>cannot be overridden</u> in subclasses?



13. **(1 bod)** For the case of a client-server architecture where there are *n* clients connected to one servers, determine the minimal number of threads on the server implemented using OCSF framework (Note: neglect all administrative threads, OS threads etc.)

14. **(1 point)** A program which takes three two-digit integer numbers as input using a keyboard has to be tested using *Combination Testing* technique. Determine the number if required test cases.

15. **(1 point)** Describe the process of incremental Top-down integration testing.

16. **(2 points)** For the procedure below:
    a. (1 point) draw the control flow graph,
    b. (1 point) determine the cyclomatic complexity.

```
public void bubbleSort(int [] array) {
int c, d, swap, n;
n = array.length;
  for (c = 0 ; c < ( n - 1 ); c++)
  {
    for (d = 0 ; d < n - c - 1; d++)
    {
      if (array[d] > array[d+1])
      {
        swap       = array[d];
        array[d]   = array[d+1];
        array[d+1] = swap;
      }
    }
  }
  return;
}
```

17. **(1 point)** Members of a software development team use GitLab version control system for their project. Where is the data which is edited by each team member stored? Is it possible for multiple members to work simultaneously on the source code?

18. **(12 points) Problem solving:** Air Quality Tracking System

Users access the Air Quality Tracking System web application via a web browser. Web application, measurements collector and measurements analyser are located on the same server computer. The collector collects measurements from multiple remote pollutant sensors using TCP/IP protocol. There are three types of sensors for: nitrogen dioxide (NO_2), carbon monoxide (CO) and the ozone (O_3). After the measurements are collected, the collector stores them in a database located on a remote computer using HTTPS protocol. Measurements analyser receives data from the collector, processes it and sends the results to the web application. Web application retrieves the data from the remote database. The communication between client (user) computer and web application is done using HTTPS protocol.

Two types of users can access the web application: clients and employees. Employees can login into the system on the main webpage. Upon the successful login, the employees are redirected to the page containing current measurements. The employees can also access the page with current warnings. On that page an employee can manually write a new warning and publish it. If the warning is already published, the employee can remove it. Simultaneously with publishing and removing warnings, the activities concerning notification of the collector and storing the changes in the database are performed. After these activities have finished, the employee is redirected to the page with current warnings, which is also visible to the clients.

In the normal mode of operation, the system collects the measurements every hour. If the measured value of one or more pollutants exceeds a given limit, the system goes into the exceptional mode of operation in which the frequency of collecting measurements is increased. In the exceptional mode of operation, two situations can occur:

- Measured value of a single type of pollutant exceeds the given limit and system goes into the accelerated mode where the measurements are collected every 15 minutes;
- Measured values of multiple pollutants exceed the limits and system goes into critical mode of operation where measurements are collected every minute.

The system automatically changes the mode of operation based on the information about the level of pollutants given by the analyser. When entering the exceptional mode of operation, web application records the time of entry and the level of all pollutants in the log which is stored in the remote database. On exiting the exceptional mode of operation, only the time of exit is recorded into the log. During the exceptional mode of operation, all users can see a note: ”Pollutant limit exceeded” on the page with current warnings, while on the page with current measurements all exceeding values are marked red. In the case of a system failure, upon recovery, the system is automatically restored to the same mode of operation in which it was before the failure.

a. (4 points) Use UML Statemachine diagram to model modes of operation for the Air Quality Tracking System.

b. (4 points) Use UML Activity diagram to model the activities related to warning publishing, including the worker login.

c. (4 points) Use UML Deployment diagram to model the entire Air Quality Tracking System on the instance level. Choose the names of nodes and components randomly.

Note for all diagrams: do not model possible system failures.