Høgskolen i Agder
Avdeling for økonomi- og samfunnsfag

EKSAMEN

Emnekode:	**IS-202**
Emnenavn:	**Programming Related Topics**
Dato:	February 28.
Varighet:	**0900-1300**
Antall sider inkl. forside:	**3**
Målform:	**English**
Tillatte hjelpemidler:	**Students are allowed to use any books or other printed material and handwritten notes.**
Merknader:	**Read the entire problem set before you start to write your answer.** **Problem 2a asks for a change to the solution to 1b. You may answer both in a single program, and use comments to point out the answer to 2a.**

Introduction

In this problem set we will look at various aspects of user registration for a public web service. You can imagine the service is a web shop, but the problems are the same for many other types of service such as forums or wikis.

Problem 1: User Registration (15%, 25%)

The simplest form of user registration is to let the users fill in a form, and use a servlet to register the users in some kind of database. You can use the User class (see appendix) to handle storing and retrieving of user information from the database.

a) Write the html code for the user registration form. The form must have input fields for the desired username, password and email address.

b) Write the user registration servlet, to handle the form.
The servlet must check the that the user have provided a username, password and email address (In other words, that all fields have been filled in) and that the username is not in use by someone else.
If everything is in order the servlet creates the new user account by calling User.createUser().
The servlet displays a page with an appropriate message.

Problem 2: User Activation (5%, 25%)

The user registration we created in problem 1 will work well enough, but we want to ensure that only humans can register (For a number of reasons, to make the service less vulnerable to spamming and denial of service attacks are two of them).
One way to do this is to send an email to the new user with an activation URL which the user must open in his browser to activate the account.

a) Modify the servlet from problem 1b, so it will call User.sendActivationEmail() to send an activation email.

b) Write an activation servlet that will handle the activation URLs from the activation emails. The activation URL is on the form:
http://www.myserver.com/activate?user=username&key=secretkey.
The value of the parameter *user* is the username, and the value of the parameter *key* is a unique key generated by the User class. Use User.activate() to activate the account and display a welcome message if the activation was successful, and an error message if it fails.

Problem 3: User security (20%, 10%)

a) People who use system infrequently may forget their passwords. We do not want them to create new accounts, so we need a way to remind them of their password, or give them a new password. Describe a solution to this problem and how you would implement it. (In this case you may propose changes to the User class as weill). You do not have write any code, just describe what you would have done.
b) Can you think of any alternatives to using passwords for authentication? Are they suitable for a web shop?

Appendix: Class User

```
/**
 * Class for storing and retrieving user information.
 * You do not need to make any changes to this class.
 * You can assume that it was written by someone else.
 */
public class User {

    /**
     * Get user information from the database, using username as a key.
     * @param username the username of the user to retrieve
     * @return the User object representing the user
     * @return null if the user does not exist
     */
    public static User getUser(String username);

    /**
     * Create a new user and store the information in the database.
     * The new user must be activated before he can log in.
     * @param username the username of the new user
     * @param password the password of the new user
     * @param email the email address of the new user
     * @return the new User object.
     * @throw IllegalArgumentException if any of the parameters is null,
     * or the username is not unique.
     */
    public static User createUser(String username, String password,
        String email) throws IllegalArgumentException;

    /** Constructor is private. The only way to create user objects
     * is to call one of the static methods above. */
    private User();

    /** Get the username */
    public String getUsername();

    /** Get the email address */
    public String getEmail();

    /** Check whether a user has been activated */
    public boolean isActivated();

    /**
     * Send an email with the activation URL to the users email address.
     * @throws Exception if the email address is invalid, or sending
     * the message fails for other reasons. */
    public void sendActivationEmail throws Exception();

    /**
     * Activate the user. The key must be the same as we sent
     * in the activation email.
     * @param key the key parameter from the activation url.
     * @return true if the key was correct and the user was activated.
     * false otherwise
     */
    public boolean activate(String key);
}
```