United International University
Department of Computer Science and Engineering

CSE 3421: Software Engineering Mid: Spring 2024

Total Marks: 30 Time: 1 hour and 30 minutes

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

Answer all the questions. Numbers to the right of the questions denote their marks.

1. **Write GIT commands** for the following tasks: (6)
   - Initialize a GIT repository
   - Set your username and email
   - Create 2 branches named UI and Chat
   - Create 2 files named **image.txt** and **text.txt**.
   - Commit **image.txt** in the master branch, and **text.txt** in Chat branch
   - Edit the **text.txt** file and commit it in Chat branch
   - Create a new file **design.txt** and commit it in UI branch
   - Show history of the file **text.txt**
   - Edit the **design.txt** file and commit it in UI branch
   - Revert back to the previous commit of UI branch (Code: abc123)
   - Merge UI, Chat branches with the master branch
   - Upload all of your work to Github

2. (a) **Define Business Requirement Specification**. How does **Functional Requirement differ** from **Non-Functional Requirement**? **Explain** with **one example**. (2)

   (b) Scientists have rescued a dying alien from the outer space. But nobody has any idea about its food habits or desirable environmental conditions. Now, they are putting it in a controlled room. The room's parameters i.e. temperature, humidity, and amount of gases are manually controlled by the scientists for now. You are given the task of developing a software that can change these parameters. But as the scientists aren't sure about different aspects of the creature, the software requirements are likely to change very frequently. On the other hand, the software needs to be sophisticated as it is dealing with the health condition of a life. Which **model** should be used to build the project? If you think one particular model can't address the necessities properly, you can use elements of different models which are relevant to this project. **Explain your choice** with **proper reasoning**. **Describe** the **working principles**. (4)

3. (a) Your company is creating an AR glass and trying to sell it in as many countries as possible. It has created two versions of the same documentation which is demonstrated in figure 1 (2)

(a) Doc-A (b) Doc-B

Figure 1: Documentation for upcoming AR glass

What **type** of documentation do these fall in? Which one should you **prefer** over the other and **describe** your reason.

   (b) You're the owner of a famous restaurant. In your database are various menu items along with their recipes and prices. Your users can only view the food items along with their prices. On top of that, each user has a profile page where they can share anything they want. (4)
   Case 1: Lately, you are noticing that your recipes are being shared on various social media platforms. It means someone has got their hands on your recipes although your system allows none but you to see those. What **kind** of **cyber-attack** might be responsible for this scenario? **Explain** the attack process.
   Case 2: Many of the users are seeing that their profiles have posted a certain recipe. But none of them have done it by themselves. It seems this rogue recipe has been shared by someone else from all the accounts. **Explain** this attack.

4. (a) What does **Scrum Master** do? How does **Stand Up Meeting in scrum method** help in the software development process? (2)

(b) Explain how **Metaphor**, **On-Site Customer** of XP help with the software development process? In which cases **Collective Code Ownership** may cause **issues** despite being so beneficial? (4)

5. (a) Give **1 reason** for **when** you should refactor your code and **when** you should **not** refactor your code. (2)

(b) **Refactor** the following code: (4)

```
class c1{
        public int a,b,ans;
        public c1(int a, int b)
        {
                this.a = a;
                this.b = b;
                System.out.println("First num = " + a);
                System.out.println("Second num = " + b);
                ans = a + b;
                System.out.println("Sum = " + ans);
                ans = a - b;
                System.out.println("Difference = " + ans);
                ans = a * b;
                System.out.println("Product = " + ans);
                ans = a / b;
                System.out.println("Quotient = " + ans);
        }
        public int f1()
        {
                ans = 1;
                for(int i=1; i ≤ b; i++)
                        ans *= a;
                return ans;
        }
}
class c2
{
        int fact;
        public int factorial(int n)
        {
                fact = 1;
                for(int i=1; i ≤ n; i++)
                        fact = fact * i;
                return fact;
        }
}
```