United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Mid-Term Exam, Trimester: Summer 2023
Course Code: CSI 321/CSE 3421, Course Title: SOFTWARE ENGINEERING
Total Marks: 30, Duration: 1 hour 45 minutes

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

(Answer All Questions)

1. (a) Explain the terms BRS and FRS. [CO1] **[2 Marks]**

(b) Suppose your team has been given a government software project. The risks of the project are very high and your team needs to handle the risks properly. The project is complex and large and requires many changes throughout its development phase. Your team has been given ample time and funds for this project. Which model is best suited for this scenario? Explain the workings of the model in detail. [CO1] **[4 Marks]**

2. (a) What do mean by "Requirement Elicitation and Analysis" in requirement engineering process? Explain briefly. [CO1] **[2 Marks]**

(b) Imagine that your software has a weird glitch. When an user enters his password, he is not allowed to enter the system, but when he enter a different random user's password, he is granted access to the system. Additionally, the user can view and alter data of some other random user when entered into the system. Which of the software security properties are violated in this scenario? Explain the properties in detail. [CO1] **[4 Marks]**

3. (a) What is load testing and stress testing? Explain the differences between these two type of testing. [CO1] **[3 Marks]**

(b) Explain the "On-Site Customer" and "Collective Code Ownership" key practices that are followed in extreme programming methodology. [CO1] **[3 Marks]**

4. (a) Write appropriate git commands to perform the following tasks: [CO2] **[6 Marks]**

- Initiate a git repository
- Consider that two files named **"f1.txt"** and **"f2.txt"** have been created. Perform commit to save only **"f1.txt"** at this stage.
- Create a branch named **"b-2"**
- Commit in **"b-2"** to save **"f2.txt"**
- Update **"f2.txt"** and commit it in **"b-2"** again
- Show the differences between the first and second commit of **"b-2"** [hash code of first

commit of **"b-2"** is **a11111**, second commit of **"b-2"** is **a22222**)
- Revert back to the first commit of **"b-2"** (Let hash code of 1st commit of **"b-2"** is **a22222**)
- Merge the branch **"b-2"** with master
- Find out all the commits that updated **"f2.txt"**
- Delete the branch **"b-2"**

5. (a) Why and when should you refactor code? [CO2] **[2 Marks]**

(b) Refactor the following code: [CO2] **[4 Marks]**

```
public class C1{
        public String studentName;
        int s_id, m, gcal,c1,c2,f;
        C1(int s_id, int c1, int c2, int f){
            this.s_id = s_id;
            this.m=m;
            f = c1+c2+f;
            System.out.print(s_id);
            System.out.print(studentName);
            System.out.print("total marks:"+f);
        }
}

class C2 {
    private String n;
    private int g;

    public void totalMarks(int c1, int c2, int f) {
        f = c1+f;
        f = f+c2;
    }
    public String Grade(int t,int c1, int c2, int f) {
        t = c1+c2+f;
        if(t>=80){
            return "A";
        }
        else if(t>=70){
            return "B";
        }
        else{
            return "C";
        }
    }
}
```

```
class Grade{
    public String Grade(int t,int c1, int c2, int f) {
        t = c1+c2+f;
        if(t>=80){
            return "A";
        }
        else if(t>=70){
            return "B";
        }
        else{
            return "C";
        }
    }
}
```