36

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

FINAL SEMESTER EXAMINATION
DURATION: 3 HOURS

WINTER SEMESTER, 2022-2023
FULL MARKS: 150

SWE 4701: Software Metrics and Process

Programmable calculators are not allowed. Do not write anything on the question paper.
Answer **all 6 (six)** questions. Figures in the right margin indicate full marks of questions whereas corresponding CO and PO are written within parentheses.

---

1. a) An empirical study is designed to analyze the impact of coding practices and the quality of software to know whether a project will be successful or not. A team of 4 members initiated the study by considering Object-Oriented Programming (OOP) and Functional Programming (FP) as coding practices. They selected bug density, code maintainability, readability, complexity, and performance bottlenecks as potential contributing factors to quality. They computed the values of those factors on some software repositories developed for the mobile platform. The type of application (e.g., game, e-commerce, etc.) was not considered. They analyzed the obtained results and concluded that coding practice and quality do not impact a project's success or failure. 15 (CO4) (PO2)

   Considering the scenario, answer the following questions.

   i. Define the empirical study as hypothesis testing.
   ii. Identify Independent and Dependent variables of the study.
   iii. Which key design issues were violated in the study? How would you fix those issues?
   iv. Mention threats to the validity of the study.

   b) Table 1 represents partially collected data from the empirical study. Do you think code readability is correlated with code complexity? 7 + 3 (CO2) (PO1)

   i. Calculate the correlation coefficient between code readability and complexity, and comment on the result.
   ii. Draw an appropriate graph to represent the relation.

Table 1: Empirical study data for Question 1.b)

Project	Code readability	Code Complexity	Project Status
A	80	20	Success
B	75	30	Success
C	85	25	Success
D	70	35	Failure
E	90	15	Success
F	65	40	Failure

2. a) Design a questionnaire comprising at least five questions to measure the impact of software developers' experience and their productivity in software development. Ensure that the questions encompass all measurement scales. 10 (CO3) (PO3)

   b) How do you measure the security of a web application? Give two examples using EAVS. 4 (CO1) (PO1)

   c) Differentiate between active and passive measurement goals with examples. 4 (CO3) (PO2)

d) Briefly explain step 9 of the Goal Question Indicator Metrics (GQIM) Framework where the business goal is to improve software quality. 7 (CO3) (PO1)

3. a) An online tax filing platform requires National Identification (NID) and Tax Identification Number (TIN) for user authentication. It validates these numbers by interfacing with national databases. Users are required to input personal info, assets, and expenses information in the system. Personal info includes fiscal year, bank name, account number, and mobile number. Assets include property, salary, house rent, and land valuation. Accommodation, food, clothing, and transportation costs are considered as expenses. The system computes payable tax amounts and generates a report. The generated report can be submitted as tax filing. After submission, the system displays status like submitted, pending, or completed. Users can search by keyword and check the status at any given time. The system has excellent performance and operational ease.

i. Considering the average complexity for data items and high complexity for technical factors, calculate Function Points (FP) for the system. 7 (CO2) (PO1)

ii. Do you think Function Point (FP) is a good metric for estimating the size of software? Give your constructive feedback on FP. 5 (CO2) (PO1)

b) Figure 1 represents a Component-based System (CBS) design where A and B are two components and node 1 to node 11 are elements of those components.

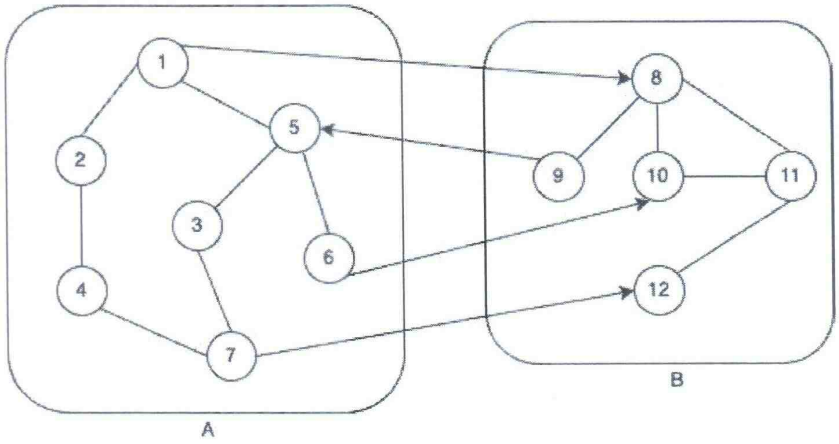

Figure 1: Component-based Software Design for Question 3.b)

i. Calculate *Cohesion* among elements of each component and the overall system. 4 (CO2) (PO1)

ii. Why do we use *Morphology*? Considering element 1 as the source node, calculate the *Morphology* for component A. 4 (CO2) (PO1)

iii. Consider that components A and B are two packages in Object Oriented Programming (OOP), and each element (i.e., 1 to 11) is a class within the package. The directed edge represents the relationship among classes across the package and the undirected edge represents the relationship within a package. Calculate *Instability* for all packages. 5 (CO2) (PO2)

4. a) If you want to propose a new complexity measure, which structural complexity properties do you need to follow? Briefly explain the properties. 5 (CO1) (PO1)

b) What is the trade-off between Cyclomatic Complexity and Data Structure Complexity? 5 (CO1) (PO1)

c) A Company runs a large software system S that has been developed in-house over a number of years. The company collects information about software defects discovered by users of S. During the regular maintenance cycle, each defect is traced to one of the 7 subsystems of S (e.g., A through G), each of which is the responsibility of a different team of programmers. Table 2 summarizes information about new defects discovered during the current year. 3 × 3 (CO2) (PO2)

Table 2: Summarized data of new defects for Question 4.c)

System	A	B	C	D	E	F	G
Defects	35	0	95	35	55	40	55
Size (KLOC)	40	100	5	50	120	70	60

Suppose you are the manager of system S. Now-

i. Compute the defect density for each subsystem.

ii. Use simple outlier analysis to identify unusual subsystems.

iii. What are the basic weaknesses of the collected metrics data?

d) The most commonly used software quality measure in industry is the number of faults per thousand lines of product source code. Compare the usefulness of this measure for developers and users. Also, mention possible problems of this measure (if any). 6 (CO2) (PO1)

5. a) Consider the Java code of Listing 1. Calculate Response For Class (RFC) and Lack of Cohesion of Methods (LCOM). Interpret the result from the design perspective. 4 + 4 (CO2) (PO1)

```
1  class GameCharacter {
2      protected String playerName;
3      protected int playerLevel;
4      protected int playerHealth;
5      public GameCharacter(String name, int level, int health) {
6          this.playerName = name;
7          this.playerLevel = level;
8          this.playerHealth = health;
9      }
10     public void attack(Enemy enemy) {
11         System.out.println(playerName + " attacks the enemy!");
12         int damage = playerLevel * 5;
13         enemy.receiveDamage(damage);
14     }
15     public void usePotion() {
16         System.out.println(playerName + " uses a health potion!");
17         int healingAmount = playerLevel * 10;
18         playerHealth += healingAmount;
19         System.out.println("Health restored by " + healingAmount);
20     }
21 }
```

Code Snippet 1: Code Snippet for Question 5.a)

b) Figure 2 represents a software lifetime where the X-axis denotes time in hours and the Y-axis denotes software is operating normally (1) or software is in the repair state (0). Calculate the Availability of the software in Figure 2, and interpret the result while the availability requirement is 77%. 5 (CO2) (PO2)

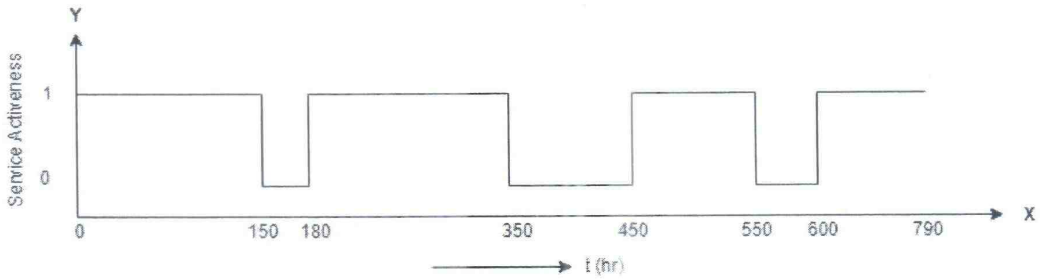

Figure 2: A software lifetime for Question 5.b)

c) What is the Reliability Model? Explain the Bathtub Curve for software reliability with a figure. 6 (CO2) (PO1)

d) A software company develops several commercial off-the-shelf (COTS) components. In normal conditions, a component operates 365 days without encountering any failure. In a study, 12 components were tested and found that 5 components failed at 250, 350, 295, 270, and 310 days. If the components' failure rate follows the Weibull Distribution with a slope parameter of 1, what is the reliability of the component at 340 days? Interpret the obtained result. 6 (CO2) (PO1)

6. Consider the primeFactors function of Listing 2.

```
1  #include <stdio.h>
2  void primeFactors(int n){
3      printf("Prime factors of %d are: ", n);
4      while (n%2 == 0){
5          printf("%d ", 2);
6          n = n/2;
7      }
8      for (int i = 3; i*i <= n; i = i+2){
9          while (n%i == 0){
10             printf("%d ", i);
11             n = n/i;
12         }
13     }
14     if (n > 2)
15         printf ("%d ", n);
16     printf ("\n");
17 }
```

Code Snippet 2: Code Snippet for Question 6.a)

a) The code prints all the prime factors of a number. For example, the prime factors of 126 are 2 3 3 7. Now, answer question 6.a). [Use appendices if necessary] 6 + 6 + 3 + 5 (CO2) (PO1)

i. Design a control flowgraph
ii. Draw a decomposition tree and calculate the depth of nesting of the flowgraph.
iii. Measure McCabe's Essential Complexity of the flowgraph.
iv. Calculate the minimum number of test cases required for branch coverage by considering the decomposition tree.

b) What is an S-structured graph? Considering $S = \{D_0, D_1\}$, what S-structured graphs can be generated? 5 (CO2) (PO1)

Appendices

Sequencing Function

Test Strategy	$F_1; ....F_n$
All-path coverage	$\prod_{i=1}^{n} \mu(F_i)$
Branch coverage	$max(\mu(F_1), ..., \mu(F_n))$
Statement coverage	$max(\mu(F_1), ..., \mu(F_n))$

Nesting Function

Test Strategy	$D_1(F_1, F_2)$	$C_n(F_1, ..., F_n)$	$D_0(F)$	$D_2(F)$
All-path coverage	$\mu(F_1) + \mu(F_2)$	$\sum_{i=1}^{n} \mu(F_i)$	$\mu(F) + 1$	-
Branch coverage	$\mu(F_1) + \mu(F_2)$	$\sum_{i=1}^{n} \mu(F_i)$	$\mu(F) + 1$	1
Statement coverage	$\mu(F_1) + \mu(F_2)$	$\sum_{i=1}^{n} \mu(F_i)$	$\mu(F)$	1

Test Strategy	$D_3(F)$	$D_4(F_1, F_2)$	$L_2(F_1, F_2$
All-path coverage	-	-	-
Branch coverage	1	1	2
Statement coverage	1	1	1

Measurement Values for Primes

Test Strategy	P_1	D_0	D_1	C_n	D_2	D_3	D_4	L_2
All-path coverage	1	2	2	n	-	-	-	-
Branch coverage	1	2	2	n	1	1	1	2
Statement coverage	1	1	2	n	1	1	1	1

Complexity weight for Function Point (FP)

Item	Low	Average	High
EI	3	4	6
EO	4	5	7
EQ	3	4	6
EIF	7	10	15
ILF	5	7	10