UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 4 Examination in Engineering: September 2023

Module Number: EE4254   **Module Name: Software Engineering Principles**

[Two Hours]

[Answer **all questions,** each question carries **10 marks**]

Instructions for candidates
- **You should submit answers to EACH QUESTION separately with a FRONT PAGE for each.**

PART II

Q1 a) Write **functional** and **nonfunctional** test scenarios to test Gmail login function.

[3.0 Marks]

   b) State the objectives of performing following test techniques.

   i)   Alpha testing

[1.0 Mark]

   ii) Regression testing

[1.0 Mark]

   iii) Smoke testing

[1.0 Mark]

   iv) Stress testing

[1.0 Mark]

   v)   Unit testing

[1.0 Mark]

   c) Explain why a system needs Ad-hoc Testing.

[2.0 Marks]

Q2 a) Name two (2) non-functional requirements related to users.

[1.0 Mark]

   b) Name two (2) requirements the team needs to gather from the clients in requirement gathering exercise?

[1.0 Mark]

   c) A company called "AnySpace" is invited to provide a business paper on how you will be delivering a business development software to a marketing company. Write down the executive summary of the report explaining your product with the requirement gathering and development strategies.
   (Expected word count: 250 words)

[8.0 Marks]

Q3 a)   Describe what is meant by 'Naming Conventions' in computer programming and why they are important.

[1.0 Mark]

b)   Describe the different types of naming conventions in Java, JavaScript and Python with suitable examples.

[2.0 Marks]

c)   Read the following statements carefully and assess if they are correct or wrong and give justifications for your answers.

i)   In clean coding, longer function names are discouraged as it makes the function harder to use and understand.

[0.5 Marks]

ii)   The Open Closed Principle in Object-Oriented Analysis and Design (OOAD) encourages modifications in the parts, only if the code has been properly tested and proper unit tests are in place.

[0.5 Marks]

iii)   While formatting code, the line length limitation of 80-100 characters per line is beneficial solely for aesthetic purposes and it doesn't impact the functionality of the code.

[0.5 Marks]

iv)   Software documentation is unnecessary if the code is self-explanatory and clean.

[0.5 Marks]