



**UGANDA INSTITUTE OF INFORMATION AND COMMUNICATIONS
TECHNOLOGY**

END OF SEMESTER ONE EXAMINATIONS

ACADEMIC YEAR 2025/2026

DEPARTMENT: ICT
YEAR OF STUDY: ONE
PROGRAMME: DIPLOMA IN SOFTWARE ENGINEERING (DSWE)
COURSE: PRINCIPLES OF PROGRAMMING
COURSE CODE: SWE 1101
DATE: MONDAY 15TH DECEMBER, 2025
TIME: 02:00 PM – 05:00 PM
DURATION: 3 HOURS

INSTRUCTIONS:

- (a) This paper consists of two sections A and B**
- (b) Section A carries 40 marks and is compulsory**
- (c) Attempt any THREE questions from section B**
- (d) Any number in section B carry equal marks**
- (e) Credit will be given for use of relevant examples and illustrations**
- (f) Begin each number on a new page of the answer sheet**

SECTION A [40 MARKS]

Attempt ALL the Questions in this section.

Subsection I:

This subsection consists of 20 multiple choice questions each for 1 mark.

List numbers vertically from 1 to 20 in the answer booklet.

Write a letter corresponding to the best alternative to each of the questions.

1. Which software development heuristic principle recommends avoiding duplication of logic?
A. KISS B. YAGNI C. DRY D. CI/CD
2. The principle “build only what is needed now, not future imagined features” refers to
A. DRY B. YAGNI C. Modularity D. TDD
3. In the SDLC, the phase that focuses on converting requirements into technical models is:
A. Maintenance B. System Design C. Testing D. Deployment
4. Which of the following is **not** a software development best practice?
A. Version control B. Refactoring
C. Code reviews D. Copy-pasting repeated code
5. The primary measure of progress in Agile is
A. Number of documents produced B. Working software
C. Lines of code D. Number of team members
6. Which Agile framework uses Sprints, Scrum Master and Product Owner?
A. Kanban B. Waterfall C. Scrum D. XP
7. In DevOps, tools like Prometheus and Grafana are mainly used for
A. Coding B. Monitoring and feedback
C. Project planning D. Deployment
8. The development practice that suggests focusing on technical excellence and simplicity is from:
A. XP B. Waterfall C. Lean D. Design Patterns
9. **Each module should handle one clear concern.** This describes
A. Testing B. Abstraction
C. Separation of Concerns D. Version control
10. Which SDLC phase involves identifying errors and verifying correctness?
A. Testing B. Deployment C. Analysis D. Maintenance
11. In DevOps, CI/CD primarily aims at
A. Delaying deployments B. Automating integration and deployment
C. Increasing documentation D. Removing testing
12. A feature of Kanban is
A. Fixed-length sprints B. WIP limits and continuous flow
C. Mandatory stand-ups D. Sprint reviews

13. Which design pattern ensures only one instance of a class exists?
A. Factory B. Observer C. Singleton D. Decorator
14. Which architectural style breaks applications into independently deployable services?
A. Monolithic B. Client-Server C. Microservices D. Layered
15. Which of the following is an advantage of modularity?
A. Tight coupling B. Harder upgrades C. Easier maintenance D. More duplication
16. A quality metric that measures stability and error-free operation is:
A. Efficiency B. Reliability C. Security D. Maintainability
17. Which DevOps concept treats infrastructure as text-based configuration?
A. Continuous Deployment B. Infrastructure as Code (IaC)
C. Pair Programming D. Lean
18. Which Git command uploads committed changes to a remote repository?
A. git add B. git commit C. git push D. git merge
19. What is the biggest advantage of using CI/CD pipelines?
A. More manual work B. Faster, more reliable releases
C. Less automation D. More documentation
20. Which Agile estimation method uses Fibonacci numbers?
A. T-shirt sizes B. Dot voting C. Planning Poker D. Bucket system

Subsection II:

This subsection consists of Scenario- based Questions.

1. Scenerio:

You have joined a software team developing a mobile payment application for a government agency. During the initial two sprints, the team is struggling with constant requirement changes from stakeholders, low collaboration between developers and users, and slow delivery of working features.

Required:

As the new team lead, explain how you would apply **the four core principles of the Agile Manifesto** to improve the project.

In your response, demonstrate:

- (a) How each principle applies to the situation.
- (b) A clear, practical action for each principle.
- (c) The expected improvement or outcome.

(10 marks)

2. Scenerio:

A public procurement system is being upgraded to include automated supplier evaluation. The project manager wants to ensure that the team correctly follows all SDLC phases to avoid the usual problems caused by rushing development (e.g., missing requirements, system bugs, delayed deployment).

Required:

Using this procurement system upgrade, describe how each **SDLC phase** would be carried out, giving **a real-life activity or deliverable** for each phase.

Your answer should clearly show how the SDLC ensures a smooth and successful project. **(10 marks)**

SECTION B [60 MARKS]

Attempt **ONLY THREE** Questions in this section.

This Section consists of **Scenario-based** and **Case Study** Questions.

Question 1

(a) Scenario:

A UICT student development team is working on a Learning Management System (LMS). At the beginning of the semester, they are unsure whether to use **Scrum** or **Kanban**. The team has:

- Several unpredictable maintenance tasks
- Some long-term features (e.g., analytics dashboard)
- A mixture of experienced and beginner developers
- A short timeline with tight academic deadlines

Required:

Based on this scenario, compare **Scrum** and **Kanban** in the areas of **planning, workflow, roles, and iteration style**, and recommend the most suitable framework for the team.

Support your recommendation with clear justification.

(b) Scenario:

You are evaluating a newly built e-requisition module for government entities. Users have complained that the system sometimes crashes, takes too long to load, and is difficult for newly appointed procurement officers to use.

Required:

- (a) Define **Software quality metrics**. (2 marks)
- (b) Explain how these three specific metrics; **reliability, maintainability, and efficiency**, can be used to assess and improve the system. (12 marks)
- (c) Give concrete examples from the scenario to show how each metric would be applied. (6 marks)

Question 2

Case Study:

UICT has partnered with the Ministry of ICT to develop a national **Online Talent Registration System**. The initial prototype is facing major issues:

- The codebase is duplicated across multiple modules
- Features are overly complicated
- Developers keep adding unnecessary features “just in case”
- Layers of the system are tightly coupled
- The system is hard to maintain and almost impossible to test
- New developers struggle to understand the structure
- Project documentation is missing or outdated

The project manager brings you in as a consultant to rescue the system before deployment.

Required:

- (a) Using examples, **discuss how the key software engineering principles (DRY, KISS, YAGNI, Separation of Concerns, Modularity, Abstraction, Reusability, Documentation, Testing)** can be applied to fix and redesign the system. (10 marks)
- (b) Explain **how these principles work together** to produce a clean, maintainable, and high-quality software product. (10 marks)

Question 3

Case Study:

A team of five developers at UICT is building a new **Hostel Management and Allocation System**.

The project is facing serious challenges:

- Developers keep overriding each other's work
- Hard to track who made which changes
- The system sometimes breaks because unstable code is pushed
- There is no proper branching strategy
- Merge conflicts are frequent and poorly resolved
- Some developers do not know how to use Git commands correctly

You have been appointed as the technical lead to introduce **proper Git workflows** and train the team.

Required:

Using examples, explain how **Git and its core commands** (such as *git clone*, *git add*, *git commit*, *git push*, *git pull*, *git branch*, *git merge* and *git checkout*) can be applied to solve the challenges in the case study.

In your answer:

- (a) Describe how Git enables collaboration, version tracking, branching, merging, and rollback. **(4 marks)**
- (b) Demonstrate how a proper Git workflow (e.g., feature branching, pull requests, code reviews) can improve the team's productivity and prevent conflicts. **(8 marks)**
- (c) Explain how Git commands would be used practically to manage contributions, trace changes, and ensure code stability. **(8 marks)**

Question 4

Case Study:

A public-sector organization is replacing its outdated procurement system with a new

Cloud-Based E-Procurement and Approval Platform.

The system must:

- Support thousands of concurrent users
- Scale during end-of-year procurement rush
- Integrate with mobile apps, payment gateways, and audit systems
- Reduce deployment downtime
- Improve development speed and reliability
- Ensure high availability and strong quality controls

The engineering manager wants a modern approach integrating:

- Agile methodologies
- DevOps culture
- CI/CD pipelines
- Cloud development environments
- Scalable architectures (layered, microservices, event-driven)
- Software quality metrics

Required:

- (a) Using your knowledge of Basic Software Engineering Principles, explain how these modern software engineering practices work together to produce scalable, maintainable, secure, and high-quality systems. **(10 marks)**
- (b) Provide examples of tools, workflows, architectures, and metrics that would fit the above e-procurement system. **(10 marks)**

Question 5

Case Study

A development team at UICT is building a **Student Academic Tracking Platform** to monitor attendance, coursework performance, and progression. The project has:

- Frequently changing academic policy requirements
- Some modules that require rapid prototyping
- Many maintenance and support tickets from users
- Several inexperienced student developers
- Limited timelines due to academic semesters

The team cannot decide which Agile methodology to adopt. Options include **Scrum, Kanban, XP, Lean, and FDD**.

Required:

- (a) Using your knowledge of Basic Software Engineering Principles, explain each of the **Agile methodologies** and compare their values, practices, and workflows. **(10 marks)**
- (b) Evaluate the scenario and recommend which method(s) or hybrid approach would be most suitable for this team.
[Provide clear justification for your recommendation] **(10 marks)**

END