



ZCAS University

**CCS3232 – SOFTWARE DESIGN AND ARCHITECTURE
MID-SEMESTER EXAMINATION**

16TH OCTOBER, 2023

16:30 HRS-19:30HRS

**TIME ALLOWED: WRITING – THREE HOURS
READING TIME – 5 MINUTES**

INSTRUCTIONS:

1. Section A: this question is compulsory and must be attempted.
2. Sections B: Answer THREE (3) questions from this section.
3. This examination paper carries a total of 100 marks.
4. Candidates must not turn this page until the invigilator tells them to do so.

SECTION A: Question 1 is compulsory and must be attempted

Question 1

A Student Management System is a software application or information system designed to efficiently and effectively manage various aspects of student-related data, processes, and interactions within an educational institution. It serves as a centralized platform to access and manage student information, academic records, and related activities.

- i) In Software Development Processes, system requirements are normally identified at an early stage. Explain and state the difference between functional and non-functional requirements of a computer system. **(4 Marks)**
- ii) Identify and briefly explain three (3) functional requirements that are associated to a Student Management System. **(6 Marks)**
- iii) Identify and briefly explain three (3) non-functional requirements that are associated to a Student Management System. **(6 Marks)**

Class Diagram, Component Diagram and Deployment Diagram are among the commonly used UML diagrams in Software Development Processes.

- iv) You are required to provide a brief general description of each of the three (3) stated UML Diagrams; clearly stating the role that is associated with each diagram. **(12 Marks)**
- v) In relation to the Student Management System, you are required to identify and explain any two (2) specific structures and / or components that are likely to be depicted in each of the three (3) UML diagrams. **(12 Marks)**

(Total: 40 Marks)

SECTION B: Attempt any THREE questions in this section

Question 2

In the context of the software architecture / design processes, constraints refer to the limitations, requirements, or conditions that impose restrictions on the software architecture and design. These constraints arise from various sources and can have a significant impact on the software design / architecture decisions and capabilities.

- i) Identify and state any four (4) constraints which could have some impact on the software design / architecture decisions and capabilities. **(4 Marks)**
- ii) Using an example, explain how each of the identified constraints could cause the associated impact on the software design / architecture decisions and capabilities. **(16 Marks)**

(Total: 20 Marks)

Question 3

Quality is a measure of excellence or the state of being free from deficiencies or defects. Quality attributes are the system properties that are separate from the functionality of the system. Implementing quality attributes makes it easier to differentiate a good system from a bad one.

- i) Quality attributes are divided into two (2) main categories: Static Quality Attributes and Dynamic Quality Attributes. Using appropriate examples, you are required to discuss the two categories; clearly stating how they differ from each other. **(8 Marks)**
- ii) The following are some of the common quality attributes: Scalability, Availability and Robustness. Using appropriate examples, you are required to explain each of the three (3) quality attributes. **(12 Marks)**

(Total: 20 Marks)

Question 4

The Object-Oriented (OO) paradigm is a programming methodology that is centered around the concept of "objects" in the development of software. The primary goal of the object-oriented paradigm is to organize code in a manner that closely mirrors the real-world entities and their relationships.

Some of the basic concepts and terminologies of object-oriented paradigm are high-lighted below: Object, Encapsulation, Abstraction, Inheritance and Polymorphism. Using appropriate examples, you are required to state the definition, purpose, and relevance associated with each of these five (5) concepts and terminologies. **(Total: 20 Marks)**

Question 5

- i) Object-Oriented Design principles provide a framework for creating software that is easier to maintain, extend, and adapt to changing requirements. Three such common principles include the **Principle of Decoupling**, the **Ensuring Cohesion**, and the **Open-Closed Principle**. You are required to discuss these three (3) principles as they relate to the development / implementation of computer systems. **(12 Marks)**
- ii) **"Design Patterns"** and **"Refactoring"** are among the common software design and architecture terminologies. You are required to discuss the two (2) terminologies. **(8 Marks)**

(8 Marks)

(Total: 20 Marks)

END OF EXAMINATION