THIS PAPER CONSISTS OF 2 SECTIONS, 14 QUESTIONS
Section A contains 10 short-answer questions, each is worth 2 marks (Total = 20 marks).
Section B contains 4 questions (Total = 40 marks).
Total: 60 marks.

CANDIDATES SHOULD ANSWER ALL QUESTIONS

Answers for Section A are to be written in the spaces provided on the exam paper.
Answers for Section B are to be written in the answer booklet provided.
Exam papers are to be collected with the examination answer booklets.

SECTION A

QUESTION 1 [2 marks]

Explain how an agile software process differs from a waterfall software process and give an advantage and a disadvantage of each.

QUESTION 2 [2 marks]

Draw a UML class diagram to capture the following situation: “Every student is enrolled in a course. Each student may be enrolled in a set of units. Some units are core units for one or more courses and some units are elective units for one or more courses.”

QUESTION 3 **[2 marks]**

Identify the **actors** and the objects in the following scenario to register a patient in a hospital management system:

The administrator enters the patient’s name, address, date of birth and emergency contact details into the system. If the patient has only public health insurance, the administrator enters the patient’s medicare number, and the system verifies this with government health database. If the patient also has private health insurance, then the administrator enters also the patient’s private health insurance details, and the system verifies these details with the private health insurance system. When these details are verified as correct, the system saves the patient's details and confirms the registration.

QUESTION 4 **[2 marks]**

Refer to Question 3 and sketch a UML sequence diagram for the scenario where the administrator registers a patient who only has public health insurance.

QUESTION 5 **[2 marks]**

Describe the **layered** software architecture, and give one advantage and one disadvantage of it.

QUESTION 6 **[2 marks]**

Describe the Façade design pattern. Give an example to show when it would be suitable to use this pattern.

QUESTION 7 **[2 marks]**

Rewrite the following as a structured rationale argument: “The main issue is what mathematics library to use for the *Warehouse-Stock* subsystem. We could use the NAG library which supports multiple programming languages including C, C++, Java, and Fortran. However, no one in the project team is familiar with this library. So a 2 weeks’ learning period will be necessary to learn the library. From a previous project, some team members have used ALGLIB, which is an open source numerical analysis library that may be used with C++, C#, and VBA. From past experience, ALGLIB is known to be not reliable at times. As *Warehouse-Stock* needs to interface with other subsystems already implemented in Java and to avoid the reliability problem with the ALGLIB, the project team decided to allocate learning time and use the NAG library.”

QUESTION 8 **[2 marks]**

A security light system has a switch and a motion sensor attached. It can be either *armed* or *unarmed*. If the switch is in the *off* position, the light is off and the system is unarmed. When the switch is turned on, the light stays off but the system is armed. If the system is armed and the motion sensor detects movement, the light comes on. If no movement is detected for 5 seconds, the light goes off.

Draw a UML state-chart diagram to describe the states of the security light system.

QUESTION 9 **[2 marks]**

Briefly describe what should be stored in a traceability table.

QUESTION 10 **[2 marks]**

Give three reasons why requirements negotiation is needed in software engineering.

SECTION B

Answers for Section B are to be written in the examination answer booklets.

All Questions in Section B refer to the *Remote Pair Programming System:*
Pair programming is an agile software development technique in which two programmers work together at one work station. One types in code while the other reviews each line of code as it is typed in. The person typing is called the driver. The person reviewing the code is called the observer. The two programmers switch roles frequently (possibly every 30 minutes or less).

Suppose that you are asked to build a system that allows *Remote Pair Programming*. That is, the system should allow the driver and the observer to be in remote locations, but both can view a single desktop in real-time. The driver should be able to edit code and the observer should be able to "point" to objects on the driver's desktop. In addition, there should be a video chat facility to allow the programmers to communicate. The system should allow the programmers to easily swap roles and record rationale in the form of video chats. In addition, the driver should be able to issue the system to backup old work.

QUESTION 11 [12 marks]

a) (4 marks) Draw a use case diagram to show all the functionality of the system.

b) (4 marks) Describe in detail four non-functional requirements for the system.

c) (4 marks) Give a prioritized list of design constraints for the system and justify your list and the ordering.

QUESTION 12 [12 marks]

a) (6 marks) Describe a software architecture that would be suitable for the system.

b) (6 marks) Present a structured rationale argument for your software architecture using the design constraints that you identify in Question 11 c) above.

QUESTION 13 [12 marks]

a) (7 marks) Propose a set of classes that could be used in your system and present them in a class diagram. **Note:** You only need to include entity objects in your class diagram.

b) (5 marks) Propose a subsystem decomposition for these classes and comment on the coupling and cohesion within this decomposition.

QUESTION 14 [4 marks]

Identify two design patterns that would be suitable for the system. Briefly explain your answer.

----------------------- END OF PAPER -----------------------