Answer All Questions in Section A and ONE question from Section B

Use separate Answerbooks for EACH section

This is a CLOSED book examination

Note: Do not answer more than the required number of questions. Clearly cross out anything you do not wish to be marked.

The use of electronic calculators is NOT permitted.
Section A
Answer All Questions

a) In object-oriented development, the concept of class has three different meanings. Briefly describe each meaning and its context. What is an object? (2 marks)

b) What are OOA, OOD and OOP? What are two important aspects of OOD? (2 marks)

c) In OOD, what is structural modelling? Which UML diagram or diagrams are used to represent structural models? (2 marks)

d) In OOD, what is behavioural or dynamic modelling? Which UML diagram or diagrams are used to represent behavioural models? (2 marks)

e) In OOD, what is the purpose of an object interaction diagram? What are the two key concepts of an object interaction diagram? In UML, what are the two alternative representations of an object interaction diagram? (2 marks)

f) What is the purpose of testing? Why is testing important? (2 marks)

g) Describe how object-oriented development impacts testing. What are the different types of class tests? (2 marks)

h) What is MVC architecture? Describe the three major components in MVC architecture. (2 marks)

i) What are the five steps for identifying packages and building package diagrams? (2 marks)

j) What are the abstract syntax and concrete syntax of a software application? Why is it important to separate these two types of syntax in the application development? (2 marks)
B1. A retail shop has a network of pay stations for processing sale and handling returns. The pay stations are connected to a mainframe computer which has access to other systems, such as accounting and banking systems. For the scope of this question, you can assume that each pay station is a standalone computer system that provides all the functions required for handling returns. Each station is operated by a human cashier. The following is a use case scenario for handling returns at a pay station.

<table>
<thead>
<tr>
<th>Use Case Scenario: Handle Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>A customer arrives at a pay station with five items to return. The cashier enters the identifier of each return item into the pay station and the pay station records the details of each return item.</td>
</tr>
<tr>
<td>After all the items have been entered into the pay station, the pay station will perform the following refund operations: If the items were paid by cash, the pay station will instruct the cashier to refund the customer by cash and print a receipt. If the items were paid by card, the customer will be asked to insert the card into the pay station and enter the password. If the password is correct, the pay station will make the refund transaction and print a receipt.</td>
</tr>
<tr>
<td>At the end of the refund, the pay station will update its return inventory.</td>
</tr>
</tbody>
</table>

a) In the above use case scenario, what are the external actors to the pay station and the domain objects? (4 marks)

b) Draw a system sequence diagram (SSD) for the above use case scenario to show the interactions between the external actors and the pay station. (8 marks)

c) Draw a design sequence diagram that realises the system event “enterRnItem”. (2 marks)

d) Draw a design sequence diagram that realises the system event “makeCashRefund”. (2 marks)

e) Draw a design class diagram based on the design sequence diagrams in c) and d). (4 marks)
B2. a) Answer the following questions.

i) What do the dotted lines and arrows indicate in the package diagram of an appointments system (shown below)? (6 Marks)
(Question B2 continues from the previous page)

ii) Which two classes in the diagram in B2. a) i) above represent the Object Persistence Classes? (2 Marks)

iii) How are the Object Persistence Classes separated from the Problem Domain Classes (Patient, and Appt.) in the diagram in B2. a) i) above? (2 Marks)

b) Write pseudo-code to specify the “simple sequential file update” algorithm in which a sequential masterfile’s records are updated by records in a sequential transaction file to produce a new masterfile containing updated records. Assume that there is a maximum of 1 transaction file record per masterfile record. (10 Marks)