Please PRINT CLEARLY your name and student number:

Name: 

Student #: 

This examination contains 12 pages (including this cover page) and 12 questions. You are responsible for ensuring that your copy of the examination is complete. Bring any discrepancy to the attention of your invigilator.

Special Instructions:

1. Put your student number and initials on the top of every page, in case the pages become separated.

2. Write your solution to each problem in the space provided. If you need more space, use the back of the pages; clearly indicate when you have done this. The invigilator(s) will supply you with scrap paper if needed.

3. If you need to make an assumption, you are urged to state it clearly and then proceed with your answer.

4. The burden of communication is upon you. Solutions not properly explained will not be considered correct. Part of proper communication is the appearance and layout. If we cannot “decode” what you wrote, we cannot grade it as a correct solution.

5. This is a closed-book examination. No memory aids, textbooks, calculators, pocket computers, smartphones, or tablets of any kind are allowed.

6. This paper must be returned, along with any scrap paper used. All of these documents must bear your name (or initials) and student number.

7. Communication of any kind concerning the problems and solutions of this examination is not allowed with anybody except the invigilator(s) or the instructor.

* Do not write in the table to the right.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>
1. [5 points] Software engineering and computer science are related, but distinct disciplines. Briefly describe the relationship and main differences between software engineering and computer science.

2. [6 points] The waterfall model is a simple and easy to use software engineering process model, but it is not without its limitations. Name TWO alternatives to the waterfall model. What are the advantages and disadvantages of each model when used to plan a software development project?
3. [3 points] Consider the role of requirements in the software engineering process. Identify THREE software engineering artefacts on which requirements documentation has an impact.

4. [3 points] Suggest why it is important to make a distinction between developing user requirements and developing system requirements in the requirements engineering process.
5. [3 points] Consider the following system description:

A manufacturing facility wishes to develop a system to automate the process of allocating staff to production lines. Once per week, the system needs to perform the allocation based on the skills and experience of the staff. The allocation must take into account details of staff holidays and sick leave. For the following week, the system must print the draft allocation list by 12:00PM noon on Friday. Only staff in the Production Planning department will be able to amend the automatic allocation to fine-tune the list. Once the amendments have been made, the final allocation list must be printed out by 5:00PM on Friday. The system must be able to handle allocation of 100 staff members at present, and should be capable of expansion to handle double that number in the future.

(a) [1.5 points] Identify THREE functional requirements.

(b) [1.5 points] Identify THREE non-functional requirements.

NOTE: You are not required to provide the category of each identified non-functional requirement.
6. [4 points] Identify and explain the ambiguity in each of the following requirements. Rewrite each of the requirements in the standard form to eliminate the identified ambiguity. Make any necessary assumptions.

1. We want the system to encrypt the data really fast.
2. The baggage is placed on the conveyor, but if it is damaged the system should halt.
3. Display the weather for the next 24 hours.
4. Shut off the pumps if the water level remains above 100 meters for more than 4 seconds.
7. [3 points] Provide at least THREE reasons why we use modelling in software engineering.

8. [4 points] Describe the role and purpose of an analysis class diagram in system modelling and analysis. Be sure to mention the specific types of classes that we aim to identify when developing an analysis class diagram. Briefly describe a common technique that can be used to identify candidate classes and relationships when constructing an analysis class diagram for a software system.
9. [8 points] Draw a UML use case diagram for the following system:

A real estate company sells houses and condos. The company employs a Listings Management System (LMS) to store evidence of listings and other details of business. The LMS is used by the real estate agents and managers on a daily basis. To be able to do any job with the system, any user must first have their credentials verified by the system. An agent can create a listing and then retrieve it later when needed. Agents can update listings as well. A listing update becomes complete only after a manager approves it. Managers can get listing reports from the system. They can also move a listing to an archive file or, when the company ceases to market a property, delete the listing.

NOTE: Be sure to use proper UML notation and naming conventions and include all elements of the diagram (e.g., system boundaries, actors, use cases, associations, and any «include», «extend», or generalization relationships).
10. [8 points] Draw a UML class diagram for the following system:

A real estate company sells houses and condos. The company employs a Listings Management System (LMS) to store evidence of listings and other details of business. A listing contains a listing number, property address, an image, the year it was built, the area, and the listing price. In the case of houses, customers usually want to know the number of floors, size of the lot, and basement characteristics. For condos, people are more interested in the size of storage and whether the property has a balcony.

The LMS can show listing assignments, that is, which listing is assigned to which agents at a time. A single agent manages a number of listings at a time. While a listing is usually managed by a single agent, a particular listing can be re-assigned from one agent to another. It must be recorded when an agent starts and ends managing a listing. For each agent, the LMS stores the agent’s seven-digit ID, name, address, and employment start date.

**NOTE:** Be sure to use proper UML notation and naming conventions and include all elements of the diagram (e.g., attributes, operations, types, visibilities, multiplicities, rolenames, associations, and any constraints, or aggregation, composition, or generalization relationships). Remember that you may not know all of the details required to have a complete class diagram. Include all of the details that you know. Refrain from making assumptions about any missing details.
11. [8 points] Consider an automobile drive system for a car with a cruise control feature. The following use case description describes a scenario for an *Engage Cruise Control* use case for the system. Draw a UML sequence diagram to illustrate the sequence of interactions between the main components of the automobile drive system in this scenario.

**NOTE:** Be sure to use proper UML notation and naming conventions and include all elements of the diagram (e.g., object instances, lifelines, activation boxes, messages of the correct type, and any necessary combined fragments).

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Engage Cruise Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brief Description</td>
<td>Enables a driver to start the vehicle and engage the cruise control feature.</td>
</tr>
<tr>
<td>Primary Actor</td>
<td>Driver</td>
</tr>
<tr>
<td>Secondary Actor</td>
<td>N/A</td>
</tr>
<tr>
<td>Precondition</td>
<td>Vehicle is off.</td>
</tr>
<tr>
<td>Dependency</td>
<td>N/A</td>
</tr>
<tr>
<td>Generalization</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Basic Flow**

1. Driver sets the Gearbox to *Park*.
2. Gearbox enables the ignition on the Dashboard.
3. Driver turns the key which sends a signal to the Dashboard.
4. Dashboard sends a signal to the Engine to start.
5. Driver sets the Gearbox to *Drive*.
7. Driver presses the cruise switch on the Dashboard.
8. IF the vehicle speed is greater than or equal to 40 km/h THEN
   a. Dashboard sends a signal to start the Cruise Control subsystem.
   b. Cruise Control subsystem notifies the Engine to maintain the current speed.
   c. Driver presses the brake which engages the Gearbox.
   d. Gearbox sends a signal to turn off the Cruise Control subsystem.
ELSE Dashboard notifies the Driver that the cruise control cannot be engaged.
9. Driver sets the Gearbox to *Park*.
10. Driver turns the key which sends a signal to the Dashboard.
11. Dashboard sends a signal to the Engine to shut off.

**Postcondition:** Vehicle is off.

<table>
<thead>
<tr>
<th>Specific Alternative Flows</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Alternative Flows</td>
<td>N/A</td>
</tr>
<tr>
<td>Bounded Alternative Flows</td>
<td>N/A</td>
</tr>
<tr>
<td>Special Requirements</td>
<td>N/A</td>
</tr>
</tbody>
</table>

* Draw your sequence diagram on the next page.*
* Draw your sequence diagram for Question 11 on this page.
12. [5 points] The following statechart shows the main states of an automobile drive system for a car with a cruise control feature. Assume that the events shiftP, shiftN, shiftD, shiftR each assign the state variable shift to P (park), N (neutral), D (drive), R (reverse), respectively. For example, shiftP performs the following assignment shift = P. Assume that initially, shift = P.

(a) [0.5 points] Provide a sequence of events that will put the car into reverse (from the initial state).

(b) [0.5 points] Provide a sequence of events that will get the car to accelerate using cruise control (from the initial state).

(c) [0.5 points] What is the shortest sequence of events needed to turn the engine off from the Coast state?
(d) [2.5 points] For each of the following properties, state whether the property is true or false, and explain your reasoning:

1. The car can only be started when the gear shift is set to N (neutral).
2. Cruise control is always disabled when the car is in reverse.
3. The ignition can never be turned off while the cruise control is active.
4. The gear shift is always set to D (drive) while the cruise control is active.
5. The only way to cancel the cruise control is to press the brake.

(e) [1 point] Describe any modifications needed for the given statechart to add the safety feature that the gear shift cannot be moved from P (park) to any other selection unless the driver is pressing the brake pedal.