Question 1 Short Questions [23 Marks]

1. Within the scrum methodology, what are the responsibilities of the following roles? [3]

   (a) Product owner

   Any of these: Responsible for product vision, final arbiter of requirements, reprioritises product backlog

   (b) Scrum master

   Any of these: Helps facilitate the scrum process, ensures the team works together smoothly

   (c) Development team

   Any of these: Implements software, writes documentation, engages in requirements engineering

2. Define a "commit" in the context of continuous integration [1]

   A commit refers to submitting a software configuration to the project database ("repository")

3. Name and describe the three phases of requirements engineering. [3]

   - Requirements gathering: finding out what the client wants, helps the customer to define what is required, etc.
   - Requirements analysis: Refining and modifying the gathered requirements
   - Requirements specification: Documenting the system requirements in a semi-formal or formal manner

4. In the context of unit tests, define a "test stub". When would you use a test stub? [2]

   A test stub simulates a part of a system that is called by the unit being tested. You would use it whenever a unit calls/depends on another component.

5. Define a "regression test" [1]

   A regression test is rerunning tests to ensure that previously tested software still works as expected when new changes are made.

6. What is the difference between inheritance and composition in Object-Oriented Design? [2]

   - Inheritance: an object inherits the properties of another object through class extension. "is a" relationship.
   - Composition: an object references another object as an instance variable. "has a" relationship.

7. Define "ad-hoc polymorphism". [1]

   A single name or symbol having multiple implementations depending on the arguments (method / operator overloading)

8. In Object-Oriented Design, what is meant by "information hiding"? [1]

   The ability to protect some components of the object from external entities (e.g. private variables).

9. Name the five views in the 4+1 Architectural View Model, and for each view name a UML diagram associated with that view. [5]

   - Logical View – Class Diagram, State Diagram
   - Process View – Activity Diagram, Sequence Diagram
   - Physical View – Deployment Diagram
   - Development View – Component Diagram
   - Use Case/Scenario View – Use Case Diagram

10. What is the difference between a device node and an execution environment node? [2]

    - Device Node – physical devices that run software
    - Execution Environment Node – software resource that runs on a device and can execute other software

11. In the context of a layered or multi-tiered architecture, what does it mean for a layer to be closed? [1]

    Layers can only interact with the layer above or below them

12. In the context of a microservices architecture, define a "service component". [1]

    Service components are highly specialised, isolated units that serve a single function

Question 2 Requirements [8 Marks]

Consider an online group communication tool, like Skype, Discord or Zoom.

1. Provide two user stories for the tool. You may not include logging in, logging out or registration. Ensure that the user stories are in the correct format. [4]

   Any valid example. Must be in Who-What-Why format
   e.g. As a user, I can send a text message in the chat box to a contact so that I can communicate with them via text.
   e.g. As a user, I can initiate a video call with a contact so that I can communicate with them face-to-face.

2. For the two user stories above, provide one user acceptance test each. Ensure that the acceptance tests are in the correct format. [4]

Any valid example that makes sense for the user stories above. Must be in Given-When-Then format.
e.g. Given that I am a user, when I type a message into the chat box with a contact and press the send button, then the message will be sent to the contact and will appear in the chat log.
e.g. Given that I am a user, when I press the video call button on the page for a contact, then it will initiate a video call in which my camera and voice feed are sent to the contact and vice-versa.

Question 3 Testing [8 Marks]

1. Suppose you are writing software that calculates how much tax someone owes given their annual salary as input. Suppose the following tax rules are in place:

   - If you earn R237 100 or less, your tax rate is 18%
   - If you earn between R237 101 and R370 500 (inclusive), your tax rate is R42 678 + 26% of any income above R237 100
   - If you earn between R370 501 and R512 800 (inclusive), your tax rate is R77 362 + 31% of any income above R370 500
   - If you earn more than R512 800, your tax rate is R121 475 + 36% of any income above R512 800

   (a) If you are running equivalence tests on the software above, what would the equivalence classes be? You can ignore invalid inputs (e.g. negative numbers). [2]

   - $[R0, R237100]$
   - $[R237101, R370500]$
   - $[R370501, R512800]$
   - $[R512801, \infty)$

   (b) For each equivalence class, provide an example of an input that you would use for boundary testing. [2]

   - $[R0, R237100]$ - $R0$ or $R237100$
   - $[R237101, R370500]$ - $R237100$ or $R370500$
   - $[R370501, R512800]$ - $R370501$ or $R512800$
   - $[R512801, \infty)$ - $R512801$

2. Consider the following python code.

```
def is_eligible_to_vote(age):
  can_vote = False
  if age >= 18:
    can_vote = True
  return can_vote
```

(a) Would a single test with *age = 21* give 100% statement coverage? Why or why not? [2]

Yes. The conditional statement is true and thus every statement in the function is executed during the test.

(b) Would a single test with *age = 21* give 100% edge coverage? Why or why not? [2]

No. The branch that skips the conditional statement (i.e. when age is less than 18) and returns can_vote as false is never tested.

Question 4 Object Model [4 Marks]

1. Consider the following python code.

```
class Cat:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def info(self):
        print(f"I'm a cat. My name is {self.name}. I'm {self.age} years
                                                old.")

    def make_sound(self):
        print("Meow")


class Dog:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def info(self):
        print(f"I'm a dog. My name is {self.name}. I'm {self.age} years
                                                old.")

    def make_sound(self):
        print("Bark")


cat1 = Cat("Kitty", 2.5)
dog1 = Dog("Fluffy", 4)

for animal in (cat1, dog1):
    animal.make_sound()
    animal.info()
    animal.make_sound()
```

What type of polymorphism is being displayed in the above example? Explain your answer. [2]

Duck typing, as the program treats the two classes the same because of a shared interface, even though there is no class hierarchy relationship between them (i.e. they do not both implement an Animal class, it is implicit)

2. Python supports method overriding but not method overloading. What is meant

by this statement? [2]

Methods can be overriden (i.e. same name for a method in a child class, but overrides parent's implementation), but not overloaded (i.e. two methods cannot have the same name)

Question 5 Architecture [9 Marks]

1. Consider the Model-View-Controller (MVC) architecture.

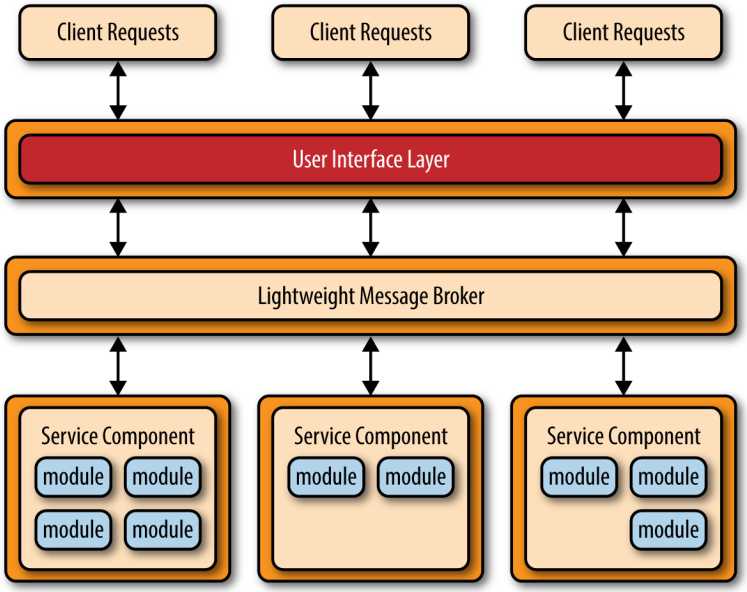
   (a) Suppose you are designing a video game and have chosen an MVC architecture for this purpose. What roles would the model, view and controller components have in this scenario? [3]

   - Model - models game world, player state, etc.
   - View - renders game world, user interface, audio, etc.
   - Controller - listens for user inputs

   (b) Name one variant of MVC architecture and describe how this variant differs from the standard MVC architecture. [2]

   Any one of the following:
   - Model-View-Presenter (MVP) - Presenter acts as intermediary between Model and View. View now responsible for user input.
   - Model-View-Viewmodel (MVVM) - Viewmodel acts as intermediary between Model and View. View binds directly to a part of the viewmodel using a binder.
   - Hierarchical MVC (HMVC) - Multiple MVC models distributed hierarchically, such that higher-level controllers can communicate with lower-level controllers.
   - Model-View-Template (MVT) - For web-based applications (specifically Django). Views handle HTTP requests and responses, while Templates handle presentation to user (e.g. html pages).

2. Consider the microservices architecture depicted below.



(a) Which variant (topology) of microservices architecture is depicted? [1]

Centralised topology

(b) What is the role of the lightweight message broker in this architecture? [1]

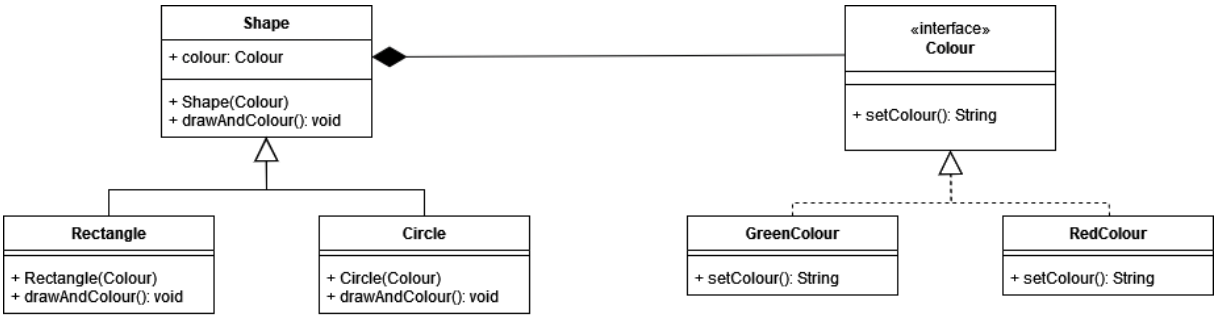
Mediates access to services, can decide which services are accessed.

(c) When designing services, it is important to consider the level of granularity of each service. We do not want services that are either too big or too small. Why is that? [2]

Too big = tends towards monolithic, lose flexibility of architecture
Too small = they may require too much central control and may rely too much on the functionality of other services, increasing coupling

Question 6 UML [8 Marks]

1. Consider the following UML diagram.



(a) What is the name of this type of UML diagram? [1]

Class diagram

(b) What is the relationship between Circle and Shape? [1]

Inheritance / class extension / generalisation

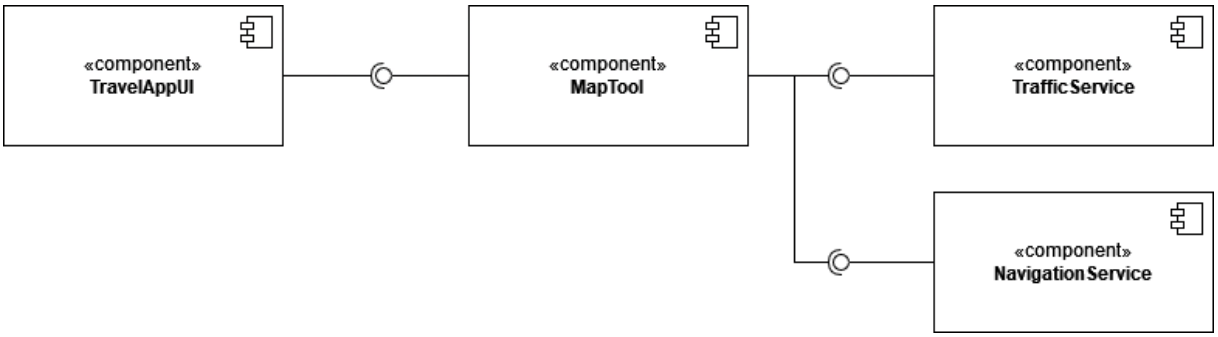
(c) What is the meaning of the relationship between Shape and Colour? Don't just give the name of the type of relationship, provide a description of what that relationship means. [1]

Composition - Shape references Colour as a variable (i.e. a Shape "has a" Colour

(d) In the diagram, there is no relationship between Circle and GreenColour. Would it still be possible to represent a green circle using this model? How? [1]

Yes. When the Circle object is instantiated, a GreenColour object can be passed to the constructor.

2. Consider the following UML component diagram.



(a) Describe the relationship between the TravelAppUI and MapTool components. [2]

MapTool provides (realises) an interface which TravelAppUI requires (uses).

(b) Suppose customers interface directly with the TravelAppUI component. What elements should be added to the diagram to represent this information? [2]

A stick figure representing the customer should be added with a dotted arrow between it and TravelAppUI representing a dependency.