CMPG223 Class Test 1 Memo

System Analysis and Design (North-West University)

CMPG223 Class Test 1 Memo - CONTACT 10 Aug 2023

Question 1 (10)

1.1. Differentiate between Aggregation and Composition in object-oriented programming. Provide a definition for each and explain the key distinctions between these two relationship concepts.
Onderskei tussen Aggregasie en Samestelling in objekgeoriënteerde programmering. Verskaf 'n definisie vir elkeen en verduidelik die sleutelverskille tussen hierdie twee verhoudingsbegrippe.

(4)

- **Aggregation:** A relationship in which one larger "whole" class contains one or more smaller "part" classes. Conversely, a smaller "part" class is part of a "whole" larger class. ✓
- **Composition:** An aggregation relationship in which the "whole" is responsible for the creation and destruction of its "parts". ✓
- **Key distinctions:** The key distinction between aggregation and composition lies in the strength of the relationship and the ownership of the parts. In aggregation, **parts can exist independently** ✓ and might be shared among different wholes. In composition, the whole class has a stronger ownership of its parts, and the parts are essential to the whole's functionality. **If the whole ceases to exist in composition, its parts will also cease to exist** ✓, which is not the case in aggregation.

1.2. Differentiate between Persistent and Transient classes in object-oriented programming.
Onderskei tussen Blywende en Kortstondige klasse in objekgeoriënteerde programmering.

(2)

- **Persistent classes are associated with objects that outlive the program's execution** and are stored permanently, while transient object classes describe objects that have a short lifespan, **existing only for the duration of the program's runtime**. ✓✓

1.3. Explain the concept of Inheritance.
Verduidelik die konsep van Oorerwing.

(1)

- The concept wherein **methods and/or attributes** defined in an object class can be **inherited or reused** by another object class. ✓

1.4. Explain the concepts of visibility in object-oriented programming with respect to attributes/methods. Define Public, Protected, and Private visibility along with the appropriate symbols to represent each visibility level.
Verduidelik die konsepte van sigbaarheid in objek-georiënteerde programmering met betrekking tot eienskappe en metodes. Definieer die openbare, beskermde en private sigbaarheid saam met die toepaslike simbole om elke sigbaarheidsvlak voor te stel.

(3)

- **Public** attributes/methods **can be accessed/invoked by any other method in any other object or class**. (Denoted by the **+** symbol) ✓
- **Protected** attributes/methods **can be accessed/ invoked by any method in the same class or in subclasses of that class**. (Denoted by the **#** symbol) ✓
- **Private** attributes/methods **can be accessed/invoked by any method in the same class**. (Denoted by the **–** symbol) ✓

Question 2 **(20)**

2.1. Verskeie stelselontwerpbenaderings word onderskei. Noem en beskryf kortliks enige TWEE stelselontwerpbenaderings wat NIE modelgedrewe is nie.
Several systems design approaches are distinguished. Name and briefly describe any TWO system design approaches that are NOT model-driven.

(4)

- **Joint Application Development (JAD)** ✓ is a technique that complements other systems analysis and design techniques by emphasizing participative development among system owners, users, designers, and builders. ✓
- **Rapid application development (RAD)** ✓– a systems design approach that utilizes structured, prototyping, and JAD techniques to quickly develop systems. ✓

2.2. Daar is vier modelgedrewe benaderings tot Stelselontwerp. Noem en bespreek drie van die benaderings.
The are four Model-Driven Approaches to System Design. Name and describe three of those approaches.

(0.5 * 6 = 3)

(Any 3):

- Modern structured design ✓ - a system design technique that decomposes the system's processes into manageable components ✓
- Information engineering ✓ - a model-driven and data-centered, but process-sensitive technique for planning, analyzing, and designing information systems. IE models are pictures that illustrate and synchronize the system's data and processes. The primary tool of IE is a data model diagram. ✓
- Prototyping ✓ - Iterative process involving a close working relationship between the designer and the users. The focus of prototyping is on a small-scale, incomplete, but working sample of a desired system ✓
- Object-oriented✓ - techniques are used to refine the object requirements definitions identified earlier during analysis, and to define design specific objects. ✓

2.3. Prototipering het baie voordele, maar ook besliste nadele. Noem enige DRIE nadele van prototipering.
Prototyping has many advantages, but also definite disadvantages. Name any THREE disadvantages of prototyping.

(3)

(Any 3):

- Encourages "code, implement, and repair" life cycle that cause maintenance nightmares. ✓
- Still need systems analysis phases, but so easy to skip. ✓
- Cannot completely substitute a prototype for a paper specification (like architect without a blueprint). ✓
- Numerous design issues are not addressed by prototyping. Often leads to premature commitment to a design.
- Scope and complexity of the system can expand out of control.
- Can reduce creativity in designs.
- Often suffer from slower performance because of language considerations (rapidly becoming a non-issue).

2.4. Noem die stelselontwerptake vir inhuis-ontwikkeling (bou).
Name the system design tasks for in-house development.

(5)

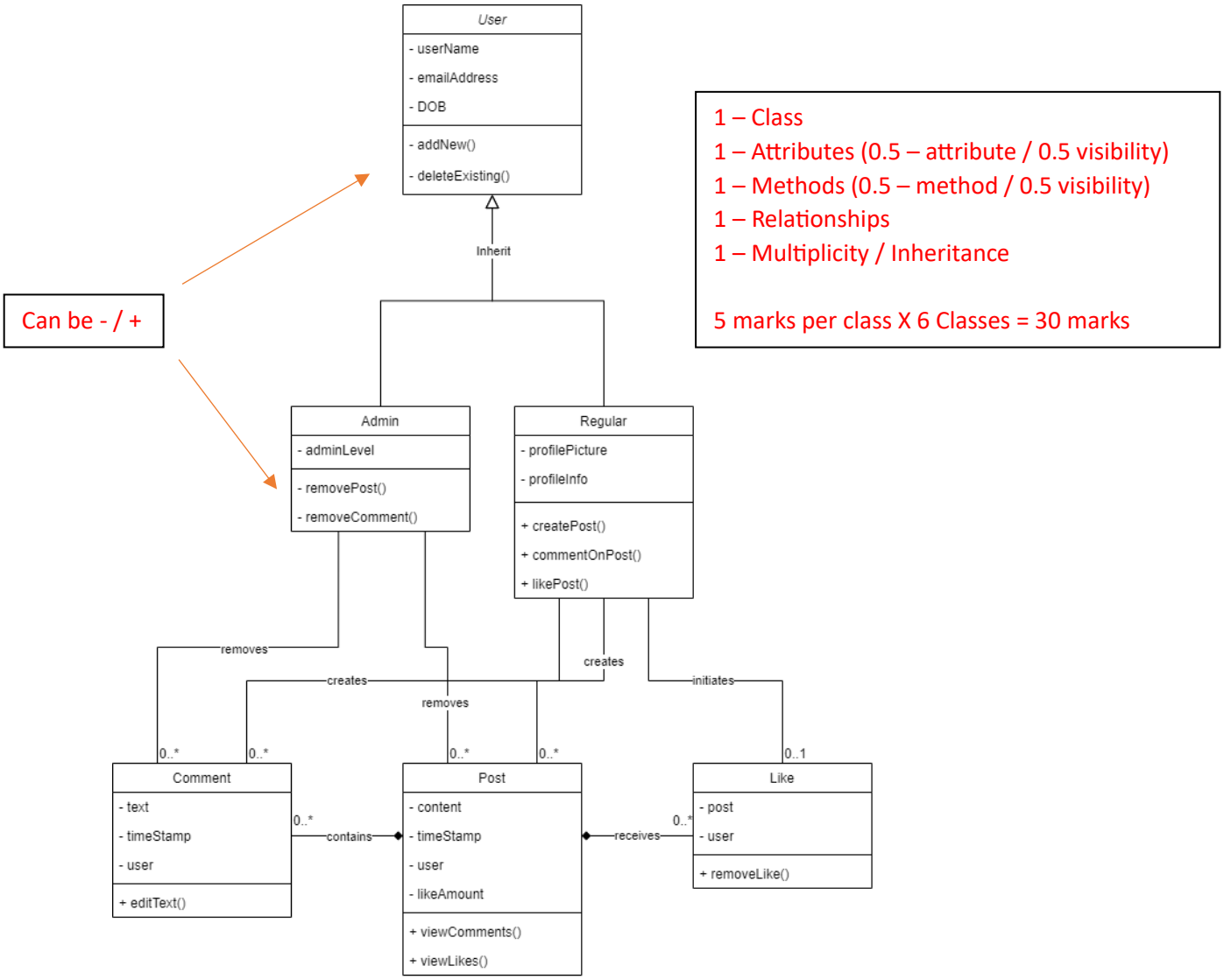
- Design the Application Architecture ✓
- Design the System Databases ✓
- Design the System Interface ✓
- Package Design Specifications ✓
- Update Project Plan ✓

2.5. Noem die take vir die verkrygingsfase van 'n kommersiële stelselsagteware oplossing.
Name the tasks for the procurement phase of a commercial systems software solution

(5)

- Research Technical Criteria and Options ✓
- Solicit Proposals or Quotes from Vendors ✓
- Validate Vendor Claims and Performances ✓
- Evaluate and Rank Vendor Proposals ✓
- Award Contract and Debrief Vendors ✓

Question 3 **(30)**



1 – Class
1 – Attributes (0.5 – attribute / 0.5 visibility)
1 – Methods (0.5 – method / 0.5 visibility)
1 – Relationships
1 – Multiplicity / Inheritance

5 marks per class X 6 Classes = 30 marks