

2022Fall-T-CSE460-70519 Exam 1

Louis Nola

TOTAL POINTS

81.5 / 100

QUESTION 1

1 Q1 5 / 8

- ✓ - 1 pts The answer to the first blank is incorrect: the correct answer "complex"
- ✓ - 1 pts The answer to the fourth blank is incorrect: the correct answer "restrained"
- ✓ - 1 pts The answer to the eighth blank is incorrect: the correct answer "discrete"

 Deducted point for each wrong answer.

✓ - 0 pts Correct

QUESTION 2

Q2 4 pts

2.1 a 2 / 2

- ✓ - 0 pts Correct

2.2 b 2 / 2

- ✓ - 0 pts Correct

QUESTION 3

8 pts

3.1 a concepts 2 / 4

- ✓ - 1 pts Concept 1 provided is partially correct
- ✓ - 1 pts Concept 2 is partially correct

 We are looking for classes and objects as one concept and hierarchy in terms of classes and not packages as 2nd concept. Partial points for abstraction or encapsulation or inheritance.. Modularity is present in 3rd gen languages as well. Hierarchy is correct but in what terms packages or class hierarchy. You should have specified that.

3.2 b benefits 4 / 4

QUESTION 4

8 pts

4.1 a 4 / 4

✓ - 0 pts Correct

4.2 b 3 / 4

✓ - 1 pts Explanation is partially correct

 The two algorithms are for one operation (i.e., multiplication). These algorithms are different in the way they multiply two numbers. For example, considering large numbers, one algorithm has high accuracy (more memory is needed) while another is less accurate (less memory is needed).

The action would result in the multiplication of two numbers and now how it's being done is hidden to the end user, who is using the calculator

Action abstraction provides a generalized set of operations all of which perform the same kind of function i.e. multiplication.

QUESTION 5

8 pts

5.1 a 7 / 8

✓ - 0 pts Correct

✓ - 1 pts Explanation of advanced principles is partially correct

 encapsulation perfect answer for that .Basic principles like encapsulation are needed to perform the calculation to hold the logic. Persistence and concurrency is for advanced

principle. A concurrency and persistence are required to handle network communication lost. Data Persistence is a means for an application to persist and retrieve information from a non-volatile storage system. A persistence unit is a complete mapping of java classes with a relational database.

QUESTION 6

16 pts

6.1 a 4 / 4

✓ - 0 pts Correct

6.2 b 6 / 6

✓ - 0 pts Correct

6.3 C 0 / 3

✓ - 3 pts microphone is a **passive** object

6.4 d 1 / 3

✓ - 2 pts An explanation is not understandable

QUESTION 7

20 pts

7.1 a 3.5 / 4

✓ - 0.5 pts State variable values should be positive finite values

7.2 b 3.5 / 4

✓ - 0.5 pts State variable values should be positive finite values

7.3 C 4 / 4

✓ - 0 pts Correct

7.4 d 6 / 8

✓ - 1 pts A pre-condition is not well-defined or well-described

✓ - 1 pts A post-condition is not well-defined or well-described

QUESTION 8

22 pts

8.1 a 12 / 12

✓ - 0 pts Correct

8.2 b 6 / 6

✓ - 0 pts Correct

8.3 C 4 / 4

✓ - 0 pts Correct

QUESTION 9

6 pts

9.1 a 2 / 2

✓ - 0 pts Correct

9.2 b 0 / 2

✓ - 2 pts Water tank class specification cannot include its own identity

9.3 C 0.5 / 2

✓ - 1.5 pts An explanation is not understandable

Midterm Exam 1: Wednesday Sep. 28, 2022

Individual Work**Format**

- Closed books and notes
- Digital media, internet access, or communication of any kind is **NOT** allowed

- Can have one two-sided, 8.5" × 11" crib sheet
- Crib sheet must be your own & include your Posting ID at top right corners
- The crib sheet must be turned in with the exam

Partial points will **NOT** be given to True/False and Fill-in-the-blank questions

Exam Coverage

All materials (including textbook chapters, course notes, homework assignments, and review sessions) covered from Aug. 22 through Sept. 27, 2022

Tempe Locations: CDN 60 and CAVC 359

Polytech Location: PRLTA 122

Classroom assignments for Tempe will be announced by 8 AM, Sept. 28, 2022

You MUST HAVE your ASU ID card to take the exam; NO other ID card is acceptable

Please arrive a few minutes early

Read questions carefully and answer what is asked for. Answer all questions.

As necessary, make appropriate assumptions & include them in your answers.

Total points: 100

NOTES:

- All specifications to be developed according to the UML standards (Astah)
- Use the Java Programming Language as needed

*** Any answer written on the last page will not be graded ***

*** Answers to questions should be written in their provided spaces ***

1. [8 points] Fill-in-blank. The items in the table below may be used never, once, or multiple times.

simple	restrained	applicable	Attributes	discrete	few	whole
abstractions	continuous	whole	parts	limitation	many	complex

Quality attributes defined for whole systems are applicable to software.

A system should have few interactions among its parts.

Software complexity is required to be Simple to achieve its user needs.

Distinct abstractions of software are necessary given limitation of human reasoning.

The hidden order property of continuous systems should be engineered for complex systems.

2. [4 points] Modularity is one of the Basic principles of the Object Model.

- (a) [2 points] Is the Modularity principle needed mainly for the behavior or structure of an object's abstraction? Mark X for either **Behavioral** or **Structural** below. Choose only one answer.

Behavioral _____ ; Structural X

- (b) [2 points] Explain your answer to Part (a).

Structural, because ~~desp~~ whether or not a software is modular, the behavior indicates its correctness. A large messy software should still have the same behaviour and functionality as a modular software.

However, if a software is modular in structure, it allows for easier better maintainability and scalability as specific parts are defined in organized structure.

3. [8 points] Consider Object-Oriented programming languages.

(a) [4 points] Name two concepts of the Object-Oriented programming languages that distinguish them from the late 3rd generation programming languages. Add your answer to the table below.

(b) [4 points] What are the benefits of the two concepts from Part (a). Provide one benefit for each concept separately. Add your answer to the table below.

	Concepts	Benefits
1	Abstraction / Hierarchy	<p>Abstraction where essential characters are grouped together are ranked in an ordering.</p> <p>Some abstractions can have an aggregation relationship and some can have a generalization & specialization relationship.</p> <p>Allows developers to focus on one aspect at a time</p>
2	Encapsulation / Modularity	<p>Lower level implementation details are hidden and now each functionality can exist as its own module where these modules can interact loosely with one another.</p> <p>Does not allow user to change / access unwanted lower level details</p>

4. [8 points] Consider a calculator that can add and multiply operations for natural numbers (positive numbers). Assume two algorithms can be used for the multiplication operations.

- (a) [4 points] What kind of abstraction (entity, action, virtual machine, coincidental) is appropriate for this calculator? ~~Entity~~ Entity

Explain your answer:

Calculator ~~is~~ is in this context is the ~~of~~ main object itself. It is not a Part-Of another object nor is it a form of another object. Hence, as it is the main object in this context, it can be abstracted as a whole unit on its own.

- (b) [4 points] What kind of abstraction (entity, action, virtual machine, coincidental) is appropriate for the multiplication operations? Action

Explain your answer:

Multiplication operations in this context is an operation that is a part of the whole calculator. It is not the entire object, but simply a functionality of an object. Hence, it can be abstracted as an action.

5. [8 points] The Object Model is defined to have two categories of principles named *Basic* and *Advanced*. Consider a digital calculator hosted on a remote computer. A user's computer may lose connection to the remote computer. Suppose the user enters the numbers needed for the calculation. Before the calculator can complete the calculation, the user computer loses connection to the remote computer. Once the connection is restored, the remote computer provides the result of the calculation to the user. Choose either **Needed** or **Not Needed** for each row and provide an explanation.

	Needed	Not Needed	Explanation
Basic principles	✓		Digital calculator has many functionalities. Each of the functionality can be packaged as one module on its own (achieve modularity). Moreover, users do not need to know the actual implementation of it (encapsulation).
Advanced principles	✓	✗	Typing is necessary when it comes to calculations. All calculations have to be strongly typed. Decimals and whole numbers should not be casted to the same type as calculation results might differ.

e.g. $2.5 \times 2.5 = 6.25 \rightarrow$ if casted as double

4

$2.5 \times 2.5 = 4 \rightarrow$ if casted as integer

6. [16 points] Consider a microphone. When it receives a sound, it converts it to electrical signals. It is used in devices such as phones.

The answers to the following question should be limited to the description provided above.

- (a) [4 points] Define one useful variable with a suitable name for the microphone.

Name 1: OnOffButton

Description 1: Variable to determine if microphone is in an on or off status, represented as type boolean (true for on, false for off)

- (b) [6 points] Define two useful functions with suitable names for the microphone. Each operation should have a name and description of what it does.

Name 1: Switch Microphone

Description 1: Toggle to turn the microphone on /off. When the state is on, it will change the boolean variable to off. When the state is off, it will change the boolean variable to on.

Name 2: Save Sound

Description 2: Functions to store the sound for future use. Can be used to save the electrical signals generated in a class attribute.

- (c) [3 points] Is the microphone a **Passive** or **Active** object? Mark X for your answer below.

Passive _____ ; Active X _____ ;

- (d) [3 points] Explain your answer for Part (c).

Microphone controls the flow of the actions. The class controls when it receives / stops receiving sound, controls what to do with the electrical signals, and converts it too. It does not depend on another class to call its functionality to run the class.

7. [20 points] Consider a water tank. It has a finite capacity. It can be filled using an inflow pipe. The tank can be emptied using an outflow pipe. The water volume in the tank should exceed some designated amount called *ReleaseVolume*. The outflow pipe is opened using a nozzle opens when the volume reaches the designated amount. Otherwise, the outflow pipe is closed. This question should be answered in terms of an **Object** defined to have state, behavior, and identity.

The answers to the following questions should be limited to the description provided above.

- (a) [4 points] Identify one useful state variable. Complete the table below.

State name	<i>CurrentCapacity</i>
Description	Determines the water capacity in the water tank at the moment
Values	<i>ReleaseVolume</i> \leq <i>CurrentCapacity</i> \leftarrow Designated Amount
Unit	<i>litres</i>

- (b) [4 points] Identify another useful state variable. Complete the table below.

State name	<i>OutflowVolume</i>
Description	Determines the volume of water flowing from the outflow pipe
Values	If <i>Current Capacity</i> $>$ <i>Designated Amount</i> , <i>Outflow Volume</i> = <i>Current Capacity</i> - <i>Designated Amount</i>
Unit	<i>litres</i>

- (c) [4 points] Identify one useful behavior that **can find something useful** about the water tank. Complete the table below.

Behavior name	Time Until Release Below Release Volume.
Behavior description	<p>How To calculate how long it takes until the water levels in the tank falls below the Release volume (designated amount).</p> <p>It can check how much longer users can use the water tank without refilling</p>

- (d) [8 points] Identify another useful behavior that **can result in a meaningful change** to the water tank. Complete the table below.

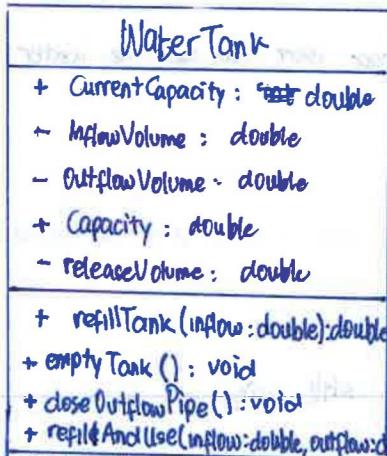
Behavior name	Refilling water tank while still using
Behavior description	<p>Behaviour which allows water tank to refill in concurrently as users still use water from the tank.</p> <p>Does not allow prevent users from still using its basic functionality while ensuring that there is still enough water.</p>
One pre-condition	<p>Inflow Volume > Outflow Volume</p> <p>To ensure that the tank is increasing in volume</p>
One post-condition	<p>Current Capacity \geq Designated Amount $\frac{\text{Release Volume}}{\text{Designated Amount}}$</p> <p>To ensure that upon refilling, the water volume exceeds the designated amount</p>

8. [22 points] Consider the water tank above. Answer the following questions according to the UML standard visual notation. The Parts (a), (b), and (c) questions should be answered separately. Don't combine the answers for the parts into one.

- (a) [12 points] Specify a class for the water tank. Name this class **WaterTank**. Include brief description for the attributes and methods.

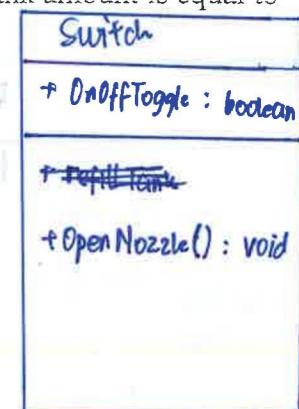
Attributes :

currentCapacity → current volume of tank
 inflowVolume → ^{Current} outflow volume
 outflowVolume → current outflow volume
 capacity → Tank Capacity
 releaseVolume → designated amount

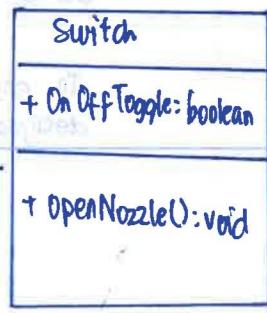
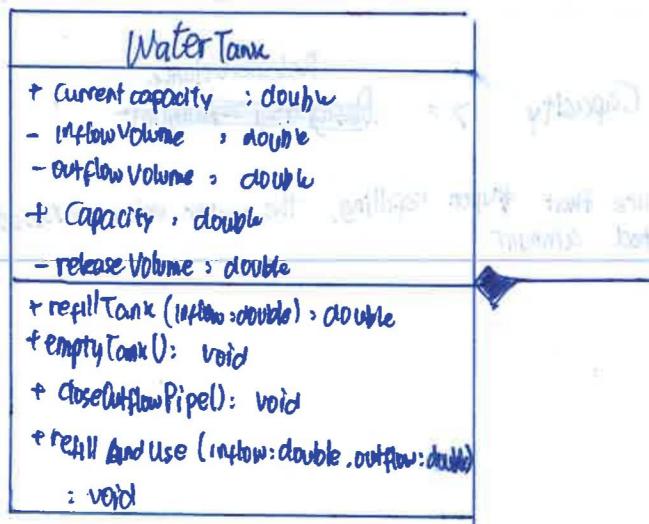
**Methods :**

refillTank → for refill tank capacity
 emptyTank → Drain all water from tank

- (b) [6 points] Consider a switch that can turn on or turn off the nozzle for the outflow pipe of the water tank. The nozzle is opened when it is notified the water tank amount is equal to *ReleaseVolume*. Specify a class for the switch.



- (c) [4 points] Specify a UML class diagram for the water tank and switch.



9. [6 points] Consider the above UML class for the water tank.

(a) [2 points] Should any object of this class (i.e., instances) be identifiable by other objects?

Yes X ; No _____

(b) [2 points] Can the water tank class specification include its own identity? Mark your answer with X.

Yes X ; No _____

(c) [2 points] Explain your answer for Part (b).

Yes, it can include its own identity. For instance,
a bigger and centralized water tank need to refill and
distribute water from to tanks around the neighborhood.
Then, the water tank should have a refill method with
its own identity specified as a parameter. Hence, its
own identity included in the class specification.

