# Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Term # \_\_\_\_

**Homework 1- UML**

**(100 points) Hours:**

**HARCOPIES - H: The homework is to be turned in as a *PAPER AND PENCIL i.e., HANDWRITTEN ANSWER ONLY* unless Textual Analysis for UML Modeling is required (also Electronic - E), in the first ten minutes of the due date class.**

**TURNING IN THE HOMEWORK INSTRUCTIONS will be PENALTY OF -10 points.**

**I UNDERSTAND THAT TURNING ANOTHER’s WORK IN is CHEATING.**

**I UNDERSTAND THAT ANY KIND OF DISSEMINATION of this WORK is CHEATING.**

**I CERTIFY THAT THE HOMEWORKs SOLUTIONs ARE MY OWN WORK!**

**?**

**X**

**V**

**SIGNATURE:**

**HOMEWORK CHECKLIST (YOU MUST GRADE YOURSELF!):**

**TA check, is Homework1.doc**

**in BB?**

1. **DID TURN IN HOMEWORK INSTRUCTIONS? \* -10 points**

**1.? 10 points**

**2.? 10 points**

**3.? 10 points**

**4.? H & E (attach to BB) – WORD ONLY 10 points**

TA checks

**5.? 10 points**

**6.? 10 points**

**7.? 15 points**

**8.? H & E (attach to BB) – WORD ONLY 15 points**

**9.? 10 points**

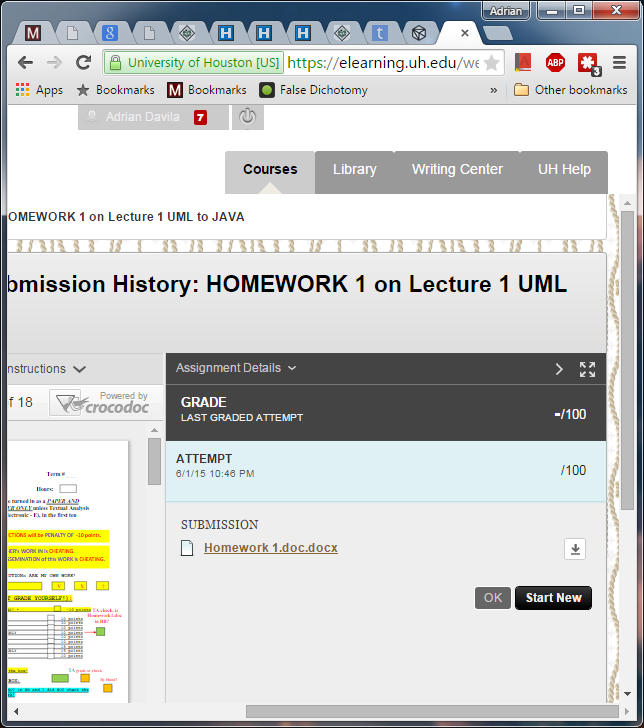
TA **grade or check**

* **If NOT, do not enter anything in the box!**

**By Hand?**

**PLEASE ENTER YOUR GRADE IN THIS BOX:**

**I understand that if the .zip file is NOT in BB and I did NOT check the BOX, I will get a ZERO for this Homework!**



**1.** (10 pts.)(**BY HAND**):

a. In a UML diagram, the top box contains the \_\_\_\_.

A: member functions, parameter list, and return type of the functions

B: name of the class

C: scope of the members

D: member variables and their data types

**ANSWER:**

b. In OOD, the first step is to identify the components, also called \_\_\_\_.

A: objects

B: functions

C: conditions

D: repetition structures

**ANSWER:**

c. In a UML diagram, the middle box contains the \_\_\_\_.

A: member functions, parameter list, and return type of the functions

B: scope of the members

C: name of the class

D: member variables and their data types

**ANSWER:**

d. Which of the following is true of a class?

A: It grows and shrinks during program execution.

B: Its components must be of the same type.

C: It contains operations, but not data.

D: It contains both data and operations.

**ANSWER:**

e. In a **UML** diagram, a(n) \_\_\_\_ sign in front of a member name indicates that the member is a public member.

A: -

B: +

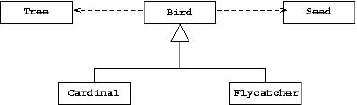
C: #

D: \*

**ANSWER:**

**2.** (10 pts.) **UML Class Diagram** (**BY HAND**):

Describe what the following **UML** Class Diagram illustrates:



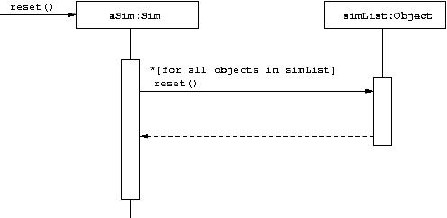
**ANSWER:**

**3.** (10 pts.) **UML Sequence Diagram** (**BY HAND**):

1. Explain the purpose of an UML sequence diagram.

**ANSWER:**

1. Describe what the following sequence UML diagram illustrates:



**ANSWER:**

**4.** (10 pts.) **UML Class Diagram** (**MICROSOFT WORD; Textual Analysis – TA Cut&Paste&Rearrange**).

Draw a **UML Class** Diagram that illustrates the following set of Classes:

A Building contains multiple Elevators, and some number of Floors. Each floor has a unique letter or number associated with it, and the floors are arranged linearly: one floor follows another in a fixed sequence. There are

passenger elevators and freight elevators. Each elevator moves from one Floor to another. Riders exist either on floors or in elevators, and there are also Freight objects. Passenger elevator have a maximum number of riders permitted, freight elevators have a limit based on total weight. Riders are assumed to have a

fixed weight, but Freight objects each have an individual weight.

**ANSWER:**

Has a

|  |
| --- |
| Building |
| Elevator Floor |
|  |

|  |
| --- |
| Floor |
| designationNumber |
|  |

|  |
| --- |
| Rider |
| WEIGHT |
| enterElevator() exitElevator() |

Has a

|  |
| --- |
| FreightElevator |
| MAX\_WEIGHT\_LIMIT |
| getFreightWeight() |

is a

|  |
| --- |
| Elevator |
| currentFloor |
| moveUp() moveDown() |

is a

uses

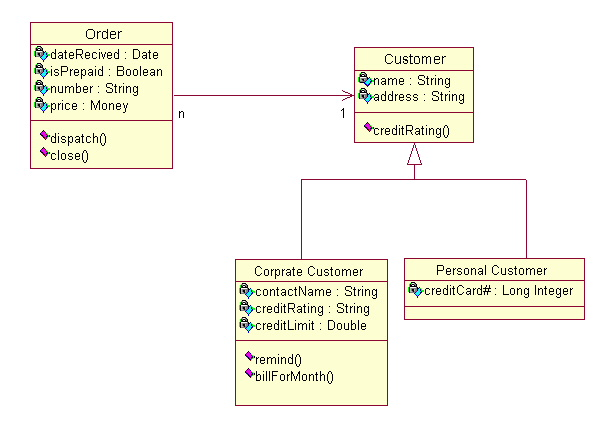
|  |
| --- |
| Freight |
| Weight |
| setFreightWeight() |

|  |
| --- |
| PassengerElevator |
| MAX\_NUMBER\_OF\_RIDERS currentRiderCount |
| getRiderCount() |

uses

**5.** (10 pts.) **UML**

Below is an **UML Class Diagram Model**



How many classes? **\_\_\_\_\_**

What are the classes’ names? **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

How many **instance variables** are in Order? **\_\_\_\_\_**

How many **methods** are in Order? **\_\_\_\_\_**

What are the relationships among the classes? **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ANSWER:**

**6.** (10 pts.) **UML Class Diagram** (**MICROSOFT WORD; Textual Analysis – TA Cut&Paste&Rearrange**):

Using the **UML graphical notation** for object classes, design the following object class identifying **attributes** and **operations**.

**A telephone**

**Answer:**

|  |
| --- |
| Telephone |
| #address  -status |
| -listenForCalls()  +connect()  +disconnect()  +makeCall() |

**7.** (15 pts.)

Based on the context specified, **identify** the most fitting **UML** relationship that best describes the relationship between each pair of **objects** or **Classes** given. Answer with one of the following: ***aggregation****,* ***association****,* ***gen/spec****,* ***instance of****.*

In addition, **DRAW** the **UML** ***Model*** that describes the relationship you identified (only the top section of the **UML** **Class *Model***).

**a. (5 pt)**

***Vending Machine*** *to* ***Vending Item***

**Answer:**

**b. (5 pt)**

***Student*** to ***Exam***

**Answer:**

**c. (5 pt)**

***Pet*** to ***Dog***

**Answer:**

**8.** (15 pts.) UML Use Case Diagram (**MICROSOFT WORD; Textual Analysis – TA Cut&Paste&Rearrange**).

**a. (5 pts)**

**Requirements Workflow** **( UML Labeled & Numbered Use Cases Diagram )**

Perform **TEXTUAL ANALYSIS** for a software **Application** that will automate a **Library** described below:

Every book in the **Library** has a bar code, a title, an author’s name, and a subject area. Every **Library** patron, called a **borrower**, has a card bearing a bar code. When a borrower wishes to *check out a book*, the **librarian** scans the bar codes on the book and the borrower’s card, and selects **C** on thesoftware **Application** Menu (or GUI). Similarly, when a *book is returned*, it is again scanned and the **librarian** enters **R** on thesoftware **Application** Menu (or GUI). **Librarians** can add or remove books (+,**-**) to the **Library** collection. **Borrowers** are only allowed to query catalog on all the books in the **Library** by a particular author (the borrower enters **A**= followed by the author’s name), all the books with a specific title (**T**= followed by the title), or all the books in a particular subject area (**S**= followed by the subject area). Finally, if a **borrower** wants a book currently checked out, the **librarian** can place a hold on the book, so that, when it is returned, it will be held for the **borrower** who requested it (**H**= followed by the number of the book).

**For Requirements Workflow UML Labeled & Numbered Use Cases Diagram**

(Mark, label and number the **Actors**; Mark, Label, and Number the **Use Cases**!)

**Controller**

**Actors: UseCase:**

**Borrower UC:CheckOutBook UC1**

**Librarian UC:ScanBookBarCode UC2**

**Librarian UC:ScanBorrowerBarCode UC3**

**Borrower UC:ReturnBook UC4**

**Librarian UC:AddBook UC5**

**Librarian UC:RemoveBook UC6**

**Borrower UC:SearchByAuthor UC7**

**Borrower UC:SearchByTitle UC8**

**Borrower UC:SearchBySubject UC9**

**Librarian UC:PlaceHold UC10**

|  |  |
| --- | --- |
| **Name:** | **CheckOutBook** |
| **Actor:** | **Borrower** |
| **Description:** | **This use case describes the process used by Borrower to check out a book** |
| **Successful Completion:** | **User requests to check out a book** |
| **Pre-Condition:** | **User requests to check out a book** |
| **Post-Condition:** | **Book checked out successfully or UNsuccessfully** |
| **Assumptions:** | **None** |

|  |  |
| --- | --- |
| **Name:** | **ScanBookBarCode** |
| **Actor:** | **Librarian** |
| **Description:** | **Librarian scans the book’s barcode** |
| **Successful Completion:** | **Book’s catalog entry is given a value of “C”.** |
| **Pre-Condition:** | **Librarian scans barcode** |
| **Post-Condition:** | **Book checked out successfully** |
| **Assumptions:** | **None** |

|  |  |
| --- | --- |
| **Name:** | **ScanBorrowerBarCode** |
| **Actor:** | **Librarian** |
| **Description:** | **Librarian scans the borrower’s barcode** |
| **Successful Completion:** | **Book’s bar code is added to Borrower’s checkout list.** |
| **Pre-Condition:** | **Librarian scans barcode** |
| **Post-Condition:** | **Book checked out successfully** |
| **Assumptions:** | **None** |
| **Name:** | **ReturnBook** |
| **Actor:** | **Librarian** |
| **Description:** | **Librarian scans the book’s barcode** |
| **Successful Completion:** | **Book’s catalog entry is given a value of “R”.** |
| **Pre-Condition:** | **Librarian scans barcode** |
| **Post-Condition:** | **Book returned successfully** |
| **Assumptions:** | **None** |

|  |  |
| --- | --- |
| **Name:** | **AddBook** |
| **Actor:** | **Librarian** |
| **Description:** | **Librarian adds book to catalog** |
| **Successful Completion:** | **Book is added to library catalog collection** |
| **Pre-Condition:** | **Book does not exist in collection** |
| **Post-Condition:** | **Book added to collection** |
| **Assumptions:** | **None** |

|  |  |
| --- | --- |
| **Name:** | **RemoveBook** |
| **Actor:** | **Librarian** |
| **Description:** | **Librarian removes book from catalog** |
| **Successful Completion:** | **Book is removed from library catalog collection** |
| **Pre-Condition:** | **Book does exist in collection** |
| **Post-Condition:** | **Book removed from collection** |
| **Assumptions:** | **None** |

|  |  |
| --- | --- |
| **Name:** | **SearchByAuthor** |
| **Actor:** | **Borrower** |
| **Description:** | **Borrower searches catalog by author** |
| **Successful Completion:** | **Displays all books by provided author** |
| **Pre-Condition:** | **Borrower presses “A=[author name]”** |
| **Post-Condition:** | **Displays all books by provided author** |
| **Assumptions:** | **None** |

|  |  |
| --- | --- |
| **Name:** | **SearchByTitle** |
| **Actor:** | **Borrower** |
| **Description:** | **Borrower searches catalog by title** |
| **Successful Completion:** | **Displays all books with provided title** |
| **Pre-Condition:** | **Borrower presses “T=[book title]”** |
| **Post-Condition:** | **Displays all books with provided title** |
| **Assumptions:** | **None** |

|  |  |
| --- | --- |
| **Name:** | **SearchBySubject** |
| **Actor:** | **Borrower** |
| **Description:** | **Borrower searches catalog by subject** |
| **Successful Completion:** | **Displays all books in provided subject** |
| **Pre-Condition:** | **Borrower presses “S=[subject]”** |
| **Post-Condition:** | **Displays all books in provided subject** |
| **Assumptions:** | **None** |

|  |  |
| --- | --- |
| **Name:** | **PlaceHold** |
| **Actor:** | **Librarian** |
| **Description:** | **Librarian places book on hold** |
| **Successful Completion:** | **Book is marked with an H and is reserved with Borrower’s bar code** |
| **Pre-Condition:** | **Book is currently checked out** |
| **Post-Condition:** | **Book is placed on hold for current Borrower** |
| **Assumptions:** | **None** |

**b. (2 pts)**

How many and who are the **Actors**?

**Answer:**

1. **(2 pts)**

How many **Use Cases** can you identify (**#**)?

**Answer:**

1. **(3 pts)**

Draw the **UML** “**COMPILABLE**” **LABELED & NUMBERED Use Cases Diagram** for the **Library Application** software product!

**Answer:**

1. **(3 pts)** Give the **Use Case Description** for the return book Use Case UC**x**:

**Answer:**

**9.** (10 pts.) **Memory**

Given the following array and pointer

char array[6];

char \*ptr;

**array[6]**

***Address Array Contents***

0x5010 a

0x5011 b

0x5012 F

0x5013 H

0x5014 X

0x5015 Y

1. Write a C assignment statement to initialize the

pointer ptr and make it point to the first element of the array?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ or\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. What are the values of the following?

&array[0]? \_\_\_\_\_ \_\_\_\_\_\_\_ (ptr+ 3)? \_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

array[0]?\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \*(ptr + 3)?\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*ptr? \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ &array[3]?\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

(\*ptr)+2?\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ \*(array+3)?\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_