

Chapter 33 OO Design Assignment

33.1)

We would modify figure 33.26 to allow a user to point to and have access to transfer a balance to another bank account. As the user currently can view their balance, withdraw cash, and deposit funds, allowing the user access here to transfer funds would be the best option in modifying the figure.

33.2)

B - Sequence Diagram. Due to the fact we want to emphasize *when* interactions are performed, we would want to have a sequence diagram as it shows the sequence in which interactions were performed. This gives us a better concept and visualization of time or “*when*” something was done.

33.3)

D - analysis, design, implementation, testing. I can confirm this from my software developer experience, it is very important to first analyze the software. If changes must be made to the software, you do not attack it head on, you first analyze the possible problem or implementation you are solving for, then design a solution for the problem or implementation. Once you have designed a change, you can then implement it, and proceed to the most important part of testing your change. Testing is by far one of the most important parts of development. I would even say during implementation, testing should be done, followed by testing after the full implementation.

33.4)

	Transmission v	
Wheels/Breaks->	Car	<-Fuel System
	^ Steering and airbags	

33.5)

C - Class computer has one-to-many relationship with class file. A computer can consist of a database of files, or folders filled with files. However, these files do not necessarily have any relationship to each other. So our class Computer would be our “one” and our multiple files and folders filled with files would be our “many.”

33.6)

True - Stated in figure 33.6, A diagram is said to be an elided diagram when some information such as the contents of the second and third compartments is *not* modeled.

33.7)

Figure 33.28 would need to be modified such that Deposit does not need access to CashDispenser, however it would need access to DepositSlot. In a withdrawal scenario, cash will be dispensed and removed from the user balance. In a scenario of depositing, instead of dispensing anything, the atm will need to accept a check, this then will add to the user balance. So very minor changes will take place in adding Deposit when comparing it to the existing withdrawal.

33.8)

B - Descriptive words and phrases. The descriptive words and phrases allow us to identify attributes that should be within our class. Just like in the ATM example, and the car example, we find words or phrases that describe attributes that we can have as part of our class, such as an account number for an ATM and a Transmission for a car.

33.9)

C - Fly, I assume they are referring to flying, flying would be an action not an attribute. So if we were to think about it in programming terms, the seats, wingspan, length, etc would be attributes, while flying would be a method.

33.10)

This attribute indicates that count is an integer with a starting value of 500. This is an attribute that keeps track of the number of bills available within the cash dispenser. If we were to declare count : Integer 1000, the cash dispenser would have a bill count of 1000.

33.11)

False - A state diagrams models some of the behavior of a system, it does not model the structural aspects of a system. So whenever an object changes state, this is what a state diagram model will show. An example is a user going from unauthenticated to authenticated.

33.12)

A - Actons. An activity diagram will model the actions that an object performs and the order in which it performs them. So if a user were to authenticate at an atm, then make a withdrawal, these actions would be modeled. An activity diagram would be like a sequence diagram in some ways when compared.

33.13)

	Prompt user to enter deposit amount or cancel
	Receive input
V-----	User canceled transaction
	Set amount attribute

	Ask user to deposit check
	Attempt to receive check
-----	Check not received
	Interact with database to add amount to users account balance
----->	end

33.14)

C - Text output is not a behavior. Reading data from a file, printing output, and obtaining input from the user are all behaviors, while text output is not. Text output is more or less referencing an actual text output. This is not the behavior of creating the text output, the behavior of creating the text output would most likely be "printing output." Text output is simply not an action

33.15)

To specify an operation that retrieves the amount attribute of classWithdrawal, the operation of getAmount() : Double would be placed in the operation compartment of class Withdrawal. As the user balance will be a double, and we are using a get method to get the amount.

33.16)

The operation add(x : Integer, y : integer) is describing an operation that takes 2 inputs, an x and a y. The x and y in this scenario would be considered parameters for the operation.

33.17)

C - A collaboration consists of an object of one class sending a message to an object of another class. You can take the word collaboration quite literally, as the object from one class is basically collaborating with an object of another class by sending it information or messages.

33.18)

Communication diagrams emphasize what collaborations occur. Just like a collaboration can be taken as a word quite literally, a communication diagram will show the communication happening from the collaboration of the objects. This is what I would call a good naming convention :) A collaboration and a communication diagram would only make sense to fit hand in hand.

33.19)

Deposit	Screen	Keypad	DepositSlot	BankDB	Account
----->	displayMessage (Message)				

----->	----->	getInput()			
----->	displayMessage(message)				
----->	----->		isEnvelopeReceived()		
----->	----->		----->	Credit (amount)---->	End

This sequence diagram models all interactions between a user and the ATM. First a message is displayed to ask the user for a deposit amount. Then the keypad awaits user input. After a message is displayed to the screen again to ask for the envelope. After, the ATM checks if it has received an envelope or not. If it has received an envelope, the bank database then credits the amount to the user account. This diagram shows the interaction between the user and the ATM, as well as the sequence at which these interactions happen in.