

SE 116 Introduction to Programming II

Lab No:	05
Topic:	POLYMORPHISM (GRADED)

You are responsible for developing a **smart AI assistant system** where different virtual assistants help users to complete various tasks. Each assistant has a **different unique role**. The system will be designed so that multiple assistants can be managed together, and users can interact with them by giving different tasks. Each concrete assistant maintains an **internal state** and will manipulate that internal state based on the user's requests.

A `VirtualAssistant` will have `assistantName` and `version` as fields. Implement its accessors and mutators and also parameterized constructor for the above fields.

There will also be `HomeAssistant` which has `isLightOn` field. Initialize it in its constructor to a default value as `false`, `PersonalFinanceAssistant` which has `currentBalance` field. Initialize it in its constructor to a default value as `500.0`, `LanguageTranslatorAssistant` which has a `lastTranslatedWord` field. Initialize it in its constructor to a default value as `"None"`. Don't forget that all of those assistant types **IS-A** `VirtualAssistant`.

(Implement their accessors and mutators for additional fields of all types of assistants.)

(Implement their parameterized constructor but don't forget we initialize additional fields to some default values!)

(Don't forget the camel case format for accessor and mutators!)

All of the virtual assistant types will have the following methods;

`String greetUser();` This method will basically greet the user however, each and every assistant type will greet differently. This method will return the greeting sentence of the assistant. The outputs will be like this:

For `HomeAssistant`:

"Hello! I'm your Home Assistant. How can I help to control your home today?"

For **PersonalFinanceAssistant**:

“Hi! I’m your Finance Assistant. Let’s manage your money wisely!”

For **LanguageTranslatorAssistant**:

“Bonjour! Hola! Hello! I’m your Language Translator AI!”

String performTask(String task); This method will get the requested task from its parameter and will perform that task based on whether the assistant is capable of doing that or not. The types of the assistants are capable of those tasks:

For **HomeAssistant**, if you asked it to “turn on lights”, it will set the `isLightOn` field to true and return “Turning on the lights!”. Also, it can turn off the lights with this command “turn off lights” and will return “Turning off the lights!”. Any other requested tasks will not be performed with this type of assistant, if that case occurred the method will return “Sorry, I can't do that.” (You need to keep track of the lights state, if it is already turned on, whenever user ask it to turn on lights, it will return “The lights are already turned on.”)

For **PersonalFinanceAssistant**, if you asked it to “show balance”, it will return you “Your current balance: 550 dollars”. If you asked it to “deposit money 100”, it will add 100 dollars into your `currentBalance` and will return “100 dollars is deposited into your account. Your current balance: 650 dollars”. Lastly, if you asked it to “withdraw 50”, if your `currentBalance` is lower than 50, the assistant will do nothing and return “Sorry, insufficient balance!” otherwise, it will withdraw 50 dollars from your `currentBalance` and will return “50 dollars is withdrawn from your account. Your current balance: 600 dollars”. Any other requested tasks will not be performed with this type of assistant, if that is the case the method will return “I don’t know how to do that.” (The deposit and withdraw values are not strict to 100 or 50. It can change. Act accordingly.)

For **LanguageTranslatorAssistant**, if you asked it to “translate hello to Spanish” it will assign `lastTranslatedWord` into “Hola” and will return “Hello in Spanish is Hola.” Also, if you asked it to “translate thank you to French” it will assign `lastTranslatedWord` to “Merci” and will return “Thank you in French is Merci.” (“Hola” and “Merci” is strict for that assistant.) Any other requested tasks will not be performed with this type of assistant, if that is the case the method will return “I don’t know that language yet.”

Finally the `AssistantManager` class, that will have the list of `VirtualAssistant` with the name `assistants` as field. Implement its accessor and mutators for that field.

Implement its default constructor and initialize the field list mentioned above to an empty list.

That class will have the following methods;

addAssistant(VirtualAssistant assistant); This method will simply add a given assistant into the `assistants` list.

removeAssistant(VirtualAssistant assistant); This method will simply remove a given assistant from the `assistants` list.

List<String> interactWithAll(String task); This method will get a requested task in its parameter and go to assistants and let them greet itself after that to ask them if they are capable of doing that task or not. After that responses(greetings and responses of they are capable or not) are gathered from those assistants and added to a list and at the end before returning the list each response is printed on the screen. Finally that response list will be returned.