**Laboratory Activity 02**

**Programming Paradigms/Object Oriented Design (PYTHON)**

1. PreLab• Readings

* 1. Python:  
     • Chap 9: Ref Lab2 Fundamentals of Python First Programs, 2nd Edition by Kenneth A. Lambert  
     • Chapter 13: Ref Lab2 Core Python Programming by Wesley J. Chun

Observation

“Fundamentals of Python First Programs” and “Core Python Programming” discussed key principles that will help someone who was starting to learn about python. Through reading these references, it can be said that python language was one of the easiest programming languages that a beginner could learn. Not only does python support object-oriented programming, but it also supports procedural programming paradigms.

In object-oriented programming, there are three important features that should be learned. These are encapsulation, inheritance, and polymorphism. These three features will assist programmers especially when maintaining applications. With encapsulation, an object’s data is accessed in a limited manner from those who uses its class’ methods. Inheritance allows one class to take the characteristic and behavior of another class. Hence the name inheritance, meaning to inherit. The subclass extends its parent class through adding data and/or methods. It can also change the methods that exists already. That is why using inheritance is a common way of reusing a code. Polymorphism on the other hand allows the use of the same headers across multiple classes. It helps programmers avoid unnecessary naming of variables by allowing the use of multiple headers while coding.

Conclusion

Python as a language is very flexible. It can be used in object-oriented programming as well as procedural programming paradigms. It is also a beginner friendly programming language. The freedom that python allows its users helps users create a code in a structure or procedure they like. The usage of different object-oriented features helps programmers have an easier time coding their programs. Unlike in java or c++, the syntax in python is also easier to remember. As a result, those who already have a background in other languages will find learning python a bit easier.

* 1. Ref Lab2 Professional Git by Brent Laster  
     • Connected Lab 2

Observation

The reference discussed Git, how to create repositories, exploring git repositories, as well as managing it. The way the reference discuss has helped in making it easier to understand these concepts. To create a repository is not a complicated thing. Just like how in computer, a folder is created to store different files, a repository is where projects can be stored. In that repository, working on the stored projects can be done. Editing, copy and pasting, and deleting can be done inside it.

Conclusion

It can be said that Git is not complicated to use. One of the reasons why a lot of different companies and programmers use it for their projects. As a student, it is a convenient way to store projects and to share projects to others without needing to pay to store said files. Therefore, the usage of it is both helpful and convenient.

* 1. UML:  
     • Chapter 4: Ref Lab2 Toolkit by Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado

Observation

The chapter four of the reference discusses UML. UML or unified modeling languages refers to a standardized modeling language that use diagrams that helps programmers in visualizing and representing objects and class. In UML the class is the most common classifier. Through a class, an object can be characterized. The objects in UML are equivalent to real software entities. Thus, designers use UML to show how a system is developed and refined. The model should always be easy to understand and coherent and a designer is tasked to make sure it is.

Primitive types are also discussed in this chapter. A primitive type is something that cannot be considered as a class and does not have any substructure. It does, however, specify a data type. In UML, primitive types include ‘string’ to represent a collection of characters while an ‘integer’ represents numbers. A ‘boolean’ however is a representation of two values, that is ‘true’ and ‘false’, and cannot represent others. Through the use of UML, linkages in an association can be indicated in different ways. UML also supports static, dynamic, and functional modeling.

Conclusion

The reference discussed how UML is used and its importance. UML provides the syntax and semantics in developing a model. Modeling language however would not tell how the job is whether it is good or bad. Models should be done in a simple to communicate, verify, validate, and manage manner so that it would not confuse those who are using it. UML also shows how rules are implemented and its restrictions thus adding complexity to the software system. Through UML, the usage of package diagrams helps organize the model to make it logical for analysis. Usage of modeling templates of classes is also possible in UML.

• Chap 5: Ref Lab 2 Systems Analysis and Design An Object-Oriented Approach with UML by Alan Dennis, Barbara Haley Wixom, David Tegarden

Observation

The book discussed about models and how it can be used for system analysis, design, and object-oriented approach. In the book, it stated that structural models are created by using iterative processes in which the model will become detailed and less conceptual. The model will depict where the data is collected such as people, places, or things. It would also show the relationships between each other. In the iterative process of building structural model, some approaches include textual analysis, brainstorming items, role playing, generating diagrams, and adding valuable patterns. These approaches help in creating a detailed model. Through these, the structural model can be understood by both the analyst and the user.

The responsibilities and collaboration of a class can also be documented. The way to do that is through using CRC cards or Class–Responsibility–Collaboration cards. These cards can compete with Unified Process’ employment of use cases and diagrams in different object-oriented systems’ development methodologies. Using these cards, a class essential elements are captured.

Conclusion

Through the use of different models, an organization’s business activities and what data structure supports it. Using different things such as class diagrams, object diagrams, and CRC card, a data structure in a system can be represented. Structural models are more logical in displaying the order of items while being analyze without displaying how these items are stored, manufactured, or even altered. These allows analysts to focus on business, avoiding technical issues. Through structural models, real world and virtual worlds are connected. It represents key things, ideas, and concepts.

• Questions and Answers

1. Although the use of a PIN to identify a person’s bank account is simple, it’s not very realistic. Real banks typically assign a unique 12-digit number to each account and use this as well as the customer’s PIN during a login at an ATM. Suggest how to rework the banking system discussed in this section to use this information.

-A six-digit PIN can be used as it is longer and thus it takes time to guess however it is not too long that the user might forget what their PIN was. Next, there should be a limit to how many tries a person can have if they entered the wrong PIN. A maximum of three tries with a pop-up message before the third try can be used. The pop-up is used to notify the user that they are on their last attempt and that once they entered the wrong PIN, the machined would not release their ATM card and locking the account associated with it for monitoring. The person could appeal to prove their identity with the bank but that is if they really own the card. These prevents others from using other peoples’ cards and guessing the PIN.

1. What is a class variable? When should the programmer define a class variable rather than an instance variable?

-Class variables are visible both inside a class definition and to external users of the class. The programmer should use class variables only for symbolic constants or to maintain data held in common by all objects of a class. For data that are owned by individual objects, programmer must use instance variables instead.

1. Describe how the arithmetic operators can be overloaded to work with a new class of numbers.

-The arithmetic operators can be overloaded to work when the operator is used to join two strings, merge two lists, and add two integers. It is possible because the operator is overloaded by the int and str classes. For example, To overload an arithmetic operator, you just define a new method using the appropriate method name, we'll need to implement the \_\_add\_\_() function in the class to overload the + operator. Great power comes with enormous responsibility. Within this function, we are free to perform whatever we choose.

2. InLab

**Objectives**

* To observe what use of visual studio and UMLet in designing program.

Converter to Kilometers to Miles Program

A screenshot of a computer

Description automatically generated

Description: The first thing to do in this code is first to make sure that the user can input a value and that value would be converted into a float value. That is why the syntax input and float are used in the first line of code. Through this, every value that the user will input will be converted into a float value. Next, a variable is created where the value used in conversion from kilometers to miles is stored. The value is inside the variable ‘convert\_val’ and is equivalent to 0.621371. Then, a variable named ‘miles’ is created. Inside of that variable, the formula for conversion is found. After the program converts the value input by the user to miles, the program will proceed to print it.

A screenshot of a computer

Description automatically generated with medium confidence

Using the extension in VS Studio Code, a UML is created. This shows the proves from user input to conversion.

**Creating diagram using Umlet in Vscode (Using Lab 2 reference by Lambert)**

Graphical user interface

Description automatically generated with medium confidence

Description: I use Umlet extension in vscode to make a diagram of page 341 in book by Lambert. The Program is The Game of BlackJack between the dealer and a player. In card game of blackjack is played with at least two players, one of whom is also a dealer. The ogject of this game is to recieve cards from the deck and play.

Using the UMLET extension in VS Studio Code a diagram is created for the game of BlackJack found on pafe 341 of the book by Lamber. In this card game, there should be at least two players, and one of those players is a dealer. The game revolves around receiving card from the game and each player’s objective is to defeat the dealer.

**Observation**

UMLET is a very convenient tool to use in creating models and diagrams. Through this models and diagrams, the flow of how the programs execute tasks is seen. These will help a lot of people even those who does not have knowledge with programming how a program works. This is important especially when dealing with clients that are not knowledgeable about coding or software design and development.

**Conclusion**

Therefore, in conclusion, it can be said that UMLET as well as UML is a very useful tool that programmers can use. It is a part of the things they use in their job especially when dealing with clients. It is important to master the usage of said tools to be more efficient in the job.

3. PostLab

Projects:

1. The ATM program allows a user an indefinite number of attempts to log in. Fix the program so that it displays a popup message that the police will be called after a user has had three successive failures. The program should also disable the login button when this happens.
2. The Doctor program described in Chapter 5 combines the data model of a doctor and the operations for handling user interaction. Restructure this program according to the model/view pattern so that these areas of responsibility are assigned to separate sets of classes. The program should include a Doctor class with an interface that allows one to obtain a greeting, a signoff message, and a reply to a patient’s string. The rest of the program, in a separate main program module, handles the user’s interactions with the Doctor object. You may develop either a terminal-based user interface or a GUI.
3. The play method in the Player class of the craps game plays an entire game without interaction with the user. Revise the Player class so that its user can make individual rolls of the dice and view the results after each roll. The Player class no longer accumulates a list of rolls but saves the string representation of each roll after it is made. Add new methods rollDice, getRollsCount, isWinner, and isLoser to the Player class. The last three methods allow the user to obtain the number of rolls and to determine whether there is a winner or a loser. The last two methods are associated with new Boolean instance variables. Two other instance variables track the number of rolls and the string representation of the most recent roll. Another instance variable tracks whether or not the first roll has occurred. At instantiation, the roll, rollsCount, atStartup, winner, and loser variables are set to their appropriate initial values. All game logic is now in the rollDice method. This method rolls the dice once, updates the state of the Player object, and returns a tuple of the values of the dice for that roll. Include in the module the playOneGame and playManyGames functions, suitably updated for the new interface to the Player class.

• Debugging and Sample Run (with screenshots and Discussion)  
• UMLET class diagram

Write your own Observation and Conclusion from what you have executed/read.

Note:1. Save all files (.py and .uxf) in one folder.

Name the folder: <Course and Section>\_LastName,FirstName\_LAB REPORT#X.  
Example: SOFTWARE DESIGN-2A\_PENTECOSTES,JAY-AR\_LAB REPORT#2.

2. Commit all Python source codes to Github **(individual Github account**)  
Github Repository Name: Software Design Lab Exercises and put a URL of your Gith