**Explanations for all python program**

**Lesson 1:**

In this lesson, I have printed simple "Hello World!" message, which is a standard introduction to any programming language. Following that, I print additional lines to get familiar with displaying messages in Python. Lastly, I use the multiplication operator to repeat a character.

**Lesson 2:**

In this program I have learned about variables. I assign different data types to variables such as integers, floats, strings, and Booleans. Then I have used the print function to output the values stored in these variables.

**Lesson 3:**

In this lesson, I used the input function to gather information from the user. The program asks for the user's name and address, and then it combines these inputs into a single message using string concatenation.

**Lesson 4:**

Here, I created a temperature converter that converts Celsius to Fahrenheit. The program collects the temperature in Celsius from the user, performs the calculation by using the equation that I have provided , and prints the result in Fahrenheit**.**

**Lesson 5:**

This lesson explores string slicing. I took a string and printed a portion of it by removing the first and last characters. String slicing allows us to extract parts of strings.

**Lesson 6:**

In this lesson, I have learned about string concatenation and formatted strings. First, I combined two variables with string concatenation, and then I used Python's formatted string literals (f-strings) to achieve the same result more cleanly.

**Lesson 7:**

This program demonstrates various string methods. I have learned by checking with different combinations with finding the length of a string, converting it to upper and lower case, searching for a substring, replacing a part of the string, and checking if a substring exists in the main string.

**Lesson 8:**

In this lesson, I explored basic arithmetic operations such as addition, subtraction, multiplication, division, modulus, and exponentiation. I also learned about compound assignment operators, which allow us to update variables in shorthand form. Finally, I demonstrated how Python handles the order of operations when performing calculations.

**Lesson 9:**

This lesson I have used math module to execute some pre built functions. I used functions like abs for absolute values, ceil and floor for rounding numbers, and factorial to calculate the factorial of a number.

**Lesson 10:**

In this lesson, I worked with conditional statements. The program checks if it’s rainy or sunny and then provides an appropriate message based on the condition. By using if, elif, and else statements, we control the flow of the program based on different logical conditions.

**Lesson 11:**

In this lesson, I learned about logical operators (and, or, not). The program evaluates whether certain conditions are true, such as the house being in good condition and reasonably priced. Depending on these conditions, it prints whether we will buy the house or if it’s a good deal.

**Lesson 12:**

Here, I explored comparison operators such as ==, !=, <, >. The program compares a price value to see if it’s equal to 10, not equal to 10, greater than 10, or less than 10.

**Lesson 13:**

This lesson introduces the while loop. I initialized a variable and printed its value until it reached 5, updating the variable within the loop. Once the condition is no longer met, the loop ends, and a final message is printed.

**Lesson 14:**

In this lesson, I worked with for loops to iterate through strings, lists, and ranges. The program demonstrates various use cases, such as looping over each character in a string, each item in a grocery list, and a range of numbers. I also calculated the total bill from a list of prices.

**Lesson 15:**

This lesson introduces nested loops. I used two for loops to generate pairs of coordinates. Each combination of x and y values from 0 to 3 is printed. This concept is useful for working with grids or matrices, where multiple dimensions are involved.

**Lesson 16:**

I worked with lists in this lesson. I printed individual items from a grocery list and used slicing to display portions of the list. Additionally, I updated an item in the list and used a loop to find the maximum value in a price list.

**Lesson 17:**

In this lesson, I explored 2D lists (matrices). I accessed and modified specific elements within the matrix and then printed all the items using nested loops.

**Lesson 18:**

This lesson covers various list methods, such as append, insert, remove, sort, reverse, and others. I also demonstrated how to check if a value exists in the list and how to count occurrences of a value. I copied the list, popped an item from the list, and cleared the list.

**Lesson 19:**

In this lesson, I worked on identifying unique numbers in a list. By checking if a number already exists in a new list, I avoid adding duplicates. This results in a list that contains only unique values.

**Lesson 20:**

Here, I learned about tuples, which are similar to lists but immutable. I printed values from a tuple and explained that unlike lists, tuples cannot be modified.

**Lesson 21:**

This lesson covers tuple unpacking and basic tuple operations. I demonstrated how to access and sum the elements of a tuple in different ways: by indexing elements individually, using unpacking into separate variables, and by directly summing the unpacked variables.

**Lesson 22:**

In this lesson, I explored dictionaries, which store data in key-value pairs. I created a dictionary to hold user information, displayed the entire dictionary, and accessed specific values using keys. I also demonstrated how to update existing values and add new key-value pairs.

**Lesson 23:**

This lesson focused on converting text to emojis. I split an input message into words, then replaced specific text symbols with corresponding emojis using a dictionary. The output is a string where symbols are converted to emojis.

**Lesson 24:**

I created functions to calculate total costs and provide a welcome message. I used keyword arguments to call the total\_cost function, demonstrating how functions can be used to perform reusable calculations. I also created a simple welcome function to illustrate how functions can encapsulate behavior and be invoked with parameters.

**Lesson 25:**

In this lesson, I defined a function to add two numbers and return the result. I then called this function with sample arguments to demonstrate how to use functions to perform specific tasks and return values.

**Lesson 26:**

I expanded on text-to-emoji conversion by encapsulating the logic in a function. The emoji\_converted function processes a message and replaces text symbols with emojis, similar to the previous lesson, but now as a reusable function.

**Lesson 27:**

This lesson involved handling exceptions with try and except blocks. I demonstrated how to manage potential errors when converting temperature input and performing calculations. By catching exceptions like ValueError and ZeroDivisionError.

**Lesson 28:**

I explained how to add comments in Python using the # symbol. Comments are used to annotate code and explain its purpose, which is crucial for code readability and maintenance.

**Lesson 29:**

I introduced classes and methods by creating a Keyboard class with methods to describe the keyboard and its number of keys. I also demonstrated how to add attributes dynamically and create multiple instances of the class.

**Lesson 30:**

This lesson covered constructors in classes using the \_\_init\_\_ method. I created a Keyboard class with attributes for language and connection type, and an AboutMe class to store personal information.

**Lesson 31:**

I demonstrated inheritance by creating a Laptop class and a Desktop class that inherits from Laptop. The Desktop class extends the functionality of the Laptop class by adding a new method.

**Lesson 32:**

I used the math module to perform mathematical operations, such as calculating the factorial of a number and displaying the value of π.

**Lesson 33:**

In this lesson, I imported functions from custom modules within a packages directory. I demonstrated how to use these functions to compute factorials, squares, and square roots.

**Lesson 35:**

I worked with the os module to interact with the file system. I showed how to get the current working directory, create and remove directories, and change directories. This lesson covers basic file and directory operations in Python, essential for managing files and directories programmatically.