**Answers to Assignment 6**

1. Python Basics:
   * What is Python, and what are some of its key features that make it popular among developers? Provide examples of use cases where Python is particularly effective.

**Answer:**

Python is a high-level programming language known for its readability and simplicity.

Features:

* Its syntax is close to natural language, makes it easy to learn and understand.
* Is powerful and versatile in that it supports various programming languages.
* Contain a wide range of libraries and frameworks.

Use cases:

* Frameworks like Django make it a powerful tool for web development.
* Libraries like NumPy makes python great for machine learning.

1. Installing Python:
   * Describe the steps to install Python on your operating system (Windows, macOS, or Linux). Include how to verify the installation and set up a virtual environment.

**Answer:**

* Download the official installer from the official Python download page (<https://www.python.org/downloads/>). Choose the latest stable version of Python
* Once downloaded, double-click the downloaded executable (.exe file).Make sure to check the box "Add Python 3.x to PATH"
* Verify it by heading to the command prompt (search for "cmd" or “Git Bash”in the Start menu). Type python --version and press Enter. If successful, you'll see the installed Python version.
* To set up the virtual environment, install the *venv or virtualenv* by running *python -m virtualenv myvirtualenv* (replace "my\_venv" with your desired environment name)
* Activate the environment by running *source virtualenv/Scripts/activate*

1. Python Syntax and Semantics:
   * Write a simple Python program that prints "Hello, World!" to the console. Explain the basic syntax elements used in the program.

**Answer:**

*print ("Hello, World!")*

* ***print ()*** is a function in python that is used to output data to the console
* String- ***“Hello, World!”***is the data to be printed on the console.

1. Data Types and Variables:
   * List and describe the basic data types in Python. Write a short script that demonstrates how to create and use variables of different data types.

**Answer:**

* **The Data Types**

1. Numerical: Like integers which represent numerical values
2. Text: Strings which represent a sequence of characters.

Example: To print age -integer

*age = 30*

*print("My age is: {age}");*

1. Control Structures:
   * Explain the use of conditional statements and loops in Python. Provide examples of an if-else statement and a for loop.

**Answer:**

Conditional statements and loops are used to control the flow of a program. Allows decision making based on conditions and iteration until certain condition is met.

***Example Else if*** *age = 18*

*if age >= 18:*

*print("You are an adult.")*

*else:*

*print("You are a child.")*

*For Loop*

*subjects = ["maths", "english", "chemistry"]*

*for subject in subjects:*

*print("I like {subject}.")*

1. Functions in Python:
   * What are functions in Python, and why are they useful? Write a Python function that takes two arguments and returns their sum. Include an example of how to call this function.

**Answers:**

* Functions are reusable blocks of code that perform specific tasks. They are useful for reusability and readability.

***Example***

*def add (x,y):*

*return x + y*

*result = add(1+2) --🡪*calling the function

*print(result)*

1. Lists and Dictionaries:
   * Describe the differences between lists and dictionaries in Python. Write a script that creates a list of numbers and a dictionary with some key-value pairs, then demonstrates basic operations on both.

**Answer:**

* Lists is the ordered collection of items while dictionary is the unordered collection of key value pairs

***Script:***

***Lists:***

*# Create a list of numbers*

*numbers = [1, 4, 5]*

*print("Numbers", numbers)*

***Dictionary with key value pairs:***

*student = {*

*"name": "Alice",*

*"age": 22,*

*"course": "Information Technology"*

*}*

*# Accessing values by key*

*name = student["name"]*

*age = student["age"]*

*course = student["course"]*

*print(f"Student Name: {name}")*

*print(f"Age: {age}")*

*print(f"Course: {course}")*

1. Exception Handling:
   * What is exception handling in Python? Provide an example of how to use try, except, and finally blocks to handle errors in a Python script.

**Answer:**

Exception handling is a mechanism in python that manages errors when executing a code.

*def divide(numerator, denominator):*

*try:*

*# Attempt the division operation*

*result = numerator / denominator*

*except ZeroDivisionError:*

*# Handle division by zero error*

*print("Error: Cannot divide by zero.")*

*result = None # Indicate an error by returning None*

*finally:*

*# Always execute this block, even if an exception occurs*

*print("Division operation completed.")*

*return result*

1. Modules and Packages:
   * Explain the concepts of modules and packages in Python. How can you import and use a module in your script? Provide an example using the math module.

**Answer:**

* Module is a single Python file containing functions, classes, variables, and statements that act as the building block of the larger program.
* Packages is the collection of related modules organized in a hierarchy

***Importing:***

*import module\_name*

***Example****:*

*import math*

*radius = 5*

*area = math.pi \* radius\*\*2*

*print(f"Area of the circle: {area:.2f}")*

1. File I/O:
   * How do you read from and write to files in Python? Write a script that reads the content of a file and prints it to the console, and another script that writes a list of strings to a file.

***Reading a file:***

*def read\_file(filename):*

***Writing a file:***

*def write\_list\_to\_file(filename, data\_list):*