To answer your assignment on Introduction to Python, I'll address each question with detailed explanations and examples where necessary.

### 1. Python Basics

**Python**: Python is a high-level, interpreted programming language known for its simplicity and readability. It supports multiple programming paradigms (procedural, object-oriented, functional) and has a vast standard library. Key features include:

* **Easy-to-learn syntax**: Simplifies coding and reduces development time.
* **Dynamic typing**: Allows flexibility by inferring types at runtime.
* **Rich ecosystem**: Extensive libraries and frameworks for various tasks.

**Use Cases**:

* **Web Development**: Django and Flask frameworks.
* **Data Analysis and Machine Learning**: NumPy, Pandas, TensorFlow.
* **Scripting**: Automation, system administration tasks.

### 2. Installing Python

**Steps**:

**Windows**:

* 1. Download Python installer from [python.org](https://www.python.org/downloads/" \t "_new).
  2. Run installer, select "Add Python to PATH".
  3. Verify installation using python --version.
  4. Set up virtual environment: python -m venv myenv.

**macOS/Linux**:

* 1. Python is pre-installed on most systems.
  2. Use package manager (e.g., brew for macOS, apt for Linux) to install if necessary.
  3. Verify installation: python3 --version.
  4. Set up virtual environment: python3 -m venv myenv.

### 3. Python Syntax and Semantics

# Simple "Hello, World!" programprint("Hello, World!")

* **Explanation**:
  + print(): Function that outputs text to the console.
  + "Hello, World!": String literal enclosed in double quotes.

### 4. Data Types and Variables

**Data Types**:

* **Integer**: x = 10
* **Float**: y = 3.14
* **String**: name = "Alice"
* **Boolean**: is\_valid = True

# Example script with variables

x = 10

y = 3.14

name = "Alice"

is\_valid = Trueprint(x, y, name, is\_valid)

### 5. Control Structures

**Conditional Statements**:

# If-else statement

age = 20if age >= 18:

print("You are an adult.")else:

print("You are a minor.")

**Loops**:

# For loopfor i in range(5):

print(i)

### 6. Functions in Python

**Functions**:

* Functions in Python are reusable blocks of code.
* Example:

# Function to add two numbersdef add\_numbers(a, b):

return a + b

# Example of function call

result = add\_numbers(3, 5)print("Sum:", result)

### 7. Lists and Dictionaries

**Lists vs Dictionaries**:

* **Lists**: Ordered collection of items accessed by index.
* **Dictionaries**: Unordered collection of key-value pairs accessed by key.

# Example of list and dictionary

my\_list = [1, 2, 3, 4, 5]

my\_dict = {"name": "Alice", "age": 30, "city": "New York"}

# Operationsprint(my\_list[0]) # Accessing list elementprint(my\_dict["name"]) # Accessing dictionary value

### 8. Exception Handling

**Exception Handling**:

* Allows handling errors gracefully.
* Example:

# Example with try, except, finally blockstry:

result = 10 / 0except ZeroDivisionError:

print("Cannot divide by zero.")finally:

print("Execution completed.")

### 9. Modules and Packages

**Modules and Packages**:

* **Modules**: Python files containing functions and variables.
* **Packages**: Directories of modules.
* Example:

# Example using math moduleimport math

print(math.sqrt(16)) # Using math function

### 10. File I/O

**File I/O**:

* **Reading from file**:

# Read content from filewith open("example.txt", "r") as file:

content = file.read()

print(content)

* **Writing to file**:

# Write list to file

data = ["apple", "banana", "cherry"]with open("fruits.txt", "w") as file:

for fruit in data:

file.write(fruit + "\n")