**Chapter 2: Python Syntax**

**Lecture Notes**

1. **Expressions, Print Statement, and Formats**

**A. Introduction to Expressions**

Expressions in Python are combinations of values, variables, and operators that produce a result when evaluated. Examples of expressions include arithmetic expressions and string concatenation.

**B. Arithmetic Operators**

Python provides various arithmetic operators such as +, -, \*, /, % (modulo), and \*\* (exponentiation) to perform mathematical operations on numeric values.

**C. String Concatenation**

String concatenation is the process of combining multiple strings into a single string. In Python, the + operator is used for string concatenation.

**D. Print Statement and Formatting**

1. **Basic Print Statement**

The print() function in Python is used to display output on the console. It can print text, variables, and expressions.

Example:

print("Hello, world!")

1. **String Formatting using %**

String formatting using the % operator allows inserting variables or values into a formatted string. Example:

name = "Alice"

age = 25

print("My name is %s and I am %d years old." % (name, age))

1. **String Formatting using f-strings**

Introduced in Python 3.6, f-strings provide a concise and readable way to format strings.

Example:

name = "Bob"

age = 30

print(f"My name is {name} and I am {age} years old.")

**2. Input Statements and Python Keywords**

**A. Reading User Input**

To obtain user input in Python, the input() function is used. It prompts the user for input and returns the entered value as a string.

**B. Handling User Input using the input() Function**

Example:

name = input("Enter your name: ")

print("Hello, " + name + "!")

**C. Python Keywords and Identifiers**

**Understanding Keywords**

Keywords in Python are reserved words that have predefined meanings and cannot be used as variable names or identifiers. Examples of keywords include if, else, while, for, etc.

**Identifiers and Naming Conventions**

Identifiers are names given to variables, functions, classes, or other objects in Python. They must follow certain naming rules and conventions. Examples of valid identifiers are age, total\_count, etc.

**3. Literals**

**A. Introduction to Literals**

Literals are constant values assigned to variables or used directly in expressions. They represent fixed values and can be of different types, such as numeric literals, string literals, boolean literals, and None literals.

**B. Numeric Literals**

**Integer Literals**

Integer literals represent whole numbers without fractional parts. Examples: 5, -10, 0.

**Floating-Point Literals**

Floating-point literals represent numbers with fractional parts. Examples: 3.14, -0.5, 1.0.

**Complex Literals**

Complex literals represent complex numbers in the form a + bj, where a and b are real numbers and j is the imaginary unit. Example: 2 + 3j.

**C. String Literals**

**Single-Quoted and Double-Quoted Strings**

Strings in Python can be defined using either single quotes ('') or double quotes (""). Examples: 'Hello', "World".

**Escape Sequences and Special Characters**

Escape sequences are used to represent special characters within strings. Examples: \n (newline), \t (tab), \ (backslash).

**D. Boolean Literals**

Boolean literals represent the truth values True and False.

**E. None Literal**

The None literal represents the absence of a value or a null value.

**4. Variables and Their Types**

**A. Variables in Python**

Variables are used to store and manipulate data in Python. They act as containers for values that can be accessed and modified during program execution.

**B. Assigning Values to Variables**

In Python, values are assigned to variables using the assignment operator (=). Example: x = 10.

**C. Variable Naming Rules and Conventions**

Python has certain rules and conventions for naming variables, such as starting with a letter or underscore, not using reserved keywords, and using lowercase letters with underscores for readability.

**D. Variable Types in Python**

Numeric Types: int, float, and complex

Numeric variables can store integer values (int), floating-point values (float), or complex numbers (complex).

String Type: str

String variables store sequences of characters.

Boolean Type: bool

Boolean variables can hold either True or False.

None Type

Variables of the None type represent the absence of a value.

**5. Comments**

**A. Purpose of Comments in Python**

Comments are used to add explanatory notes or annotations within Python code. They are ignored by the interpreter and are meant for human readers.

**B. Single-Line Comments using #**

Single-line comments start with the # symbol and extend until the end of the line.

**C. Multi-Line Comments using """ """**

Multi-line comments can be enclosed within triple quotes (""" """) and span multiple lines. They are often used for documentation purposes.