# Introduction to Programming using Python

## I. Perform Operations using Data Types and Operators

**• Evaluate an expression to identify the data type Python will assign to each variable.**

- ✓ Data types include str, int, float, and bool

**• Convert between and work with data types.**

- ✓ Type casting; constructing data structures; indexing and slicing operations

**• Determine the sequence of execution based on operator precedence.**

- ✓ Assignment; Comparison; Logical; Arithmetic; Identity (is); Containment (in)

**• Select the appropriate operator to achieve the intended result.**

- ✓ Assignment; Comparison; Logical; Arithmetic; Identity (is); Containment (in)

## II. Control Flow with Decisions and Loops

**• Construct and analyze code segments that use branching statements.**

- ✓ if; elif; else; nested and compound conditionals

**• Construct and analyze code segments that perform iteration**

- ✓ while; for; break; continue; pass; nested loops and loops that include compound conditionals

## III. Perform Input and Output Operations

**• Construct and analyze code segments that perform file input and output operations.**

- ✓ open; close; read; write; append; check existence; delete; with statement

**• Construct and analyze code segments that perform console input and output operations.**

- ✓ Read input from console; print formatted text; use of command line arguments

## IV. Document and Structure Code

• **Document code segments using comments and documentation strings.** o Use of indentation and white space; comments and documentation strings; pydoc

• **Construct and analyze code segments that include function definitions.** o Call signatures; default values; return; def; pass

## V. Perform Troubleshooting and Error Handling

• **Analyze, detect, and fix code segments that have errors.** o Syntax errors; logic errors; runtime errors

• **Analyze and construct code segments that handle exceptions.** o Try; except; else; finally; raise

## VI. Perform Operations Using Modules and Tools

• **Perform basic operations using built-in modules.**

o math; datetime; io; sys; os; os.path; random

• **Solve complex computing problems by using built-in modules.**

o math; datetime; random