Introduction to Python Programming

Course Syllabus

Course Description

This is a gentle, hands-on introduction to programming using Python. Designed for absolute beginners with no prior coding experience, the course focuses on building confidence with fundamental programming concepts through practical, engaging projects. Students will learn to think logically, solve problems step-by-step, and create working programs they can be proud of.

Learning Outcomes

By the end of this course, students will be able to:

1. Write, test, and debug simple Python programs
2. Use variables, data types, and basic operations
3. Implement decision-making with if, elif, else statements
4. Create loops with for and while
5. Work with lists, tuples, and dictionaries
6. Define and use simple functions
7. Read from and write to text files
8. Develop programs to solve everyday problems

Grading Policy

- Weekly Programming Assignments: 30%
- Lab Participation & Exercises: 20%
- Midterm Project: 20%
- Final Project: 25%
- Class Participation: 5%
- **Grading Scale:** A: 90-100%, B: 80-89%, C: 70-79%, D: 60-69%, F: Below 60%

No penalty for early struggles—focus is on improvement and final understanding.

Course Schedule

Unit 1: Getting Started (Weeks 1-2)

Week 1: Welcome to Programming

- What is programming? Why Python?
- Setting up Python and a beginner-friendly IDE
- Your first program: print("Hello, World!")
- Understanding errors and debugging
- **Lab 1:** Install Python, run first programs, explore the IDE

Week 2: Talking to the Computer

- Variables and basic data types (integers, floats, strings)
- Getting input with input()
- Basic arithmetic operations
- String concatenation and repetition
- **Lab 2:** Create a conversation program

Unit 2: Making Decisions (Weeks 3-4)

Week 3: Simple Choices

- Boolean values: True and False
- Comparison operators (==, !=, >, <, etc.)
- Simple if statements
- **Lab 3:** Create a basic quiz or age checker

Week 4: Multiple Choices

- elif for multiple conditions
- Logical operators (and, or, not)
- Nested if statements
- **Lab 4:** Build a menu-driven program (calculator, simple game)

Unit 3: Repeating Actions (Weeks 5-6)

Week 5: Counting Loops

- Introduction to for loops
- Using range() function
- Simple iteration patterns
- **Lab 5:** Create number patterns and simple counters

Week 6: Conditional Loops

- while loops
- Loop control with break and continue
- Avoiding infinite loops
- **Lab 6:** Build a number guessing game

Unit 4: Working with Data (Weeks 7-9)

Week 7: Lists – Ordered Collections

- Creating and accessing lists
- Modifying lists: append, insert, remove
- List slicing
- **Lab 7:** Create a to-do list manager

Week 8: Tuples and Dictionaries

- Tuples (immutable lists)
- Dictionaries: key-value pairs
- Basic dictionary operations
- **Lab 8:** Build a simple contact book or vocabulary trainer

Week 9: Midterm Project Week

- **Midterm Project:** Create a text-based adventure game
- Project planning and incremental development
- Peer review session

Unit 5: Organizing Code (Weeks 10-11)

Week 10: Simple Functions

- Why use functions?

- Defining functions with def
- Calling functions
- Functions with parameters
- **Lab 10:** Break programs into reusable functions

Week 11: Functions That Return Values

- The return statement
- Multiple returns
- Using functions in expressions
- **Lab 11:** Create a math helper with functions

Unit 6: Working with Files (Week 12)

Week 12: Reading and Writing Data

- Opening and closing files
- Reading from text files
- Writing to text files
- Simple data persistence
- **Lab 12:** Create a journal or note-taking program

Teaching Philosophy

1. **Mistakes are learning opportunities** – we celebrate debugging
2. **No question is too basic** – everyone starts somewhere
3. **Focus on understanding, not memorization**
4. **Learn by doing** – minimal lecture, maximum coding
5. **Build confidence through small, daily successes**

Software & Tools

- **Python 3.x** (free download)
- **Thonny IDE** (recommended for beginners) or VS Code
- **Course website** with all code examples
- **Debugging cheat sheet** provide