CoderCrafter

PYTHON PROGRAMMING SYLLABUS

Fundamentals to advanced projects

Course Overview

This syllabus provides a comprehensive overview of Python programming, designed for learners with little to no prior programming experience, as well as those looking to enhance their existing skills. The course covers fundamental concepts, object-oriented programming (OOP), web development with Flask, data processing with Pandas, automation techniques, software testing, and hands-on project development. By the end of this course, students will be proficient in Python and capable of building real-world applications.

Module 1: Python Fundamentals

- **Introduction to Python:** History, features, and applications of Python.
- **Setting up the Environment:** Installing Python and choosing an IDE (Integrated Development Environment).
- **Basic Syntax:** Variables, data types (integers, floats, strings, booleans), operators, and expressions.
- **Control Flow:** Conditional statements (, ,) and loops (,).
- **Functions:** Defining and calling functions, parameters, return values, and scope.

Module 2: Data Structures

- **Lists:** Creating, accessing, modifying, and manipulating lists.
- **Tuples:** Creating, accessing, and using tuples (immutability).
- **Dictionaries:** Creating, accessing, modifying, and using dictionaries (key-value pairs).
- **Sets:** Creating, accessing, and using sets (uniqueness).
- **List Comprehensions:** Concise syntax for creating lists.

Module 3: Object-Oriented Programming (OOP)

- **Introduction to OOP:** Concepts of classes, objects, encapsulation, inheritance, and polymorphism.
- **Classes and Objects:** Defining classes, creating objects, and accessing attributes and methods.
- **Inheritance:** Creating subclasses, inheriting attributes and methods from parent classes.
- **Polymorphism:** Method overriding and dynamic binding.
- **Encapsulation:** Hiding internal data and methods using access modifiers.

Module 4: Web Development with Flask

- **Introduction to Web Development:** Basics of HTML, CSS, and web servers.
- **Flask Framework:** Introduction to Flask, setting up a Flask application.
- **Routes and Views:** Defining routes and creating views to handle requests.
- **Templates:** Using Jinja2 templates to generate dynamic HTML.
- **Forms:** Handling user input with forms.
- **Databases:** Connecting to databases (e.g., SQLite, MySQL) using Flask-SQLAlchemy.

Module 5: Data Processing with Pandas

- **Introduction to Pandas:** Overview of Pandas library and its applications.
- **DataFrames:** Creating, accessing, and manipulating DataFrames.
- **Data Cleaning:** Handling missing data, filtering, and transforming data.
- **Data Analysis:** Performing statistical analysis, grouping, and aggregation.
- **Data Visualization:** Creating basic plots and charts using Matplotlib and Seaborn.
- **Data Import/Export:** Reading data from and writing data to various formats (e.g., CSV, Excel, JSON).

Module 6: Automation

- **File System Automation:** Automating file and directory operations.
- **Web Scraping:** Extracting data from websites using libraries like Beautiful Soup.
- **Task Scheduling:** Automating tasks using cron jobs or Task Scheduler.
- **Email Automation:** Sending and receiving emails programmatically.
- **System Administration:** Automating common system administration tasks.

Module 7: Testing

- **Introduction to Testing:** Importance of testing in software development.
- **Unit Testing:** Writing unit tests using the framework.
- **Test-Driven Development (TDD):** Writing tests before writing code.
- **Mocking:** Using mock objects to isolate code during testing.
- **Integration Testing:** Testing the interaction between different modules.

- **Debugging:** Using debugging tools to identify and fix errors.

Module 8: Hands-on Projects

Throughout the course, students will work on several hands-on projects to apply their knowledge and build practical skills. Example projects include:

- **Simple Web Application:** A basic web application with user authentication and data storage.
- **Data Analysis Project:** Analyzing a real-world dataset and generating insights.
- **Automation Script:** Automating a repetitive task, such as data backup or report generation.
- **Game Development:** Simple text-based game or using Pygame.

These projects will provide students with valuable experience in developing complete Python applications from start to finish.

Assessment

The final grade will be determined by the following:

- **Assignments (30%):** Regular assignments to reinforce concepts learned in each module.
- **Midterm Exam (20%):** A comprehensive exam covering the first half of the course.
- **Final Project (40%):** A significant project demonstrating the student's ability to apply their knowledge to solve a real-world problem.
- **Participation (10%):** Active participation in class discussions and online forums.

Resources

- **Textbook:** *Python Crash Course* by Eric Matthes
- **Online Resources:** Python documentation, Stack Overflow, GitHub.
- **IDE:** VSCode or PyCharm

Next Steps

Upon completion of this course, students are encouraged to explore more advanced topics in Python, such as machine learning, data science, or specific web frameworks like Django. Continued practice and project development are essential for mastering Python and building a successful career in software development.