🚀 Join our Intermediate Python Programming Course with Abdullah Hanif, a passionate student pursuing a Master’s in Data Science! 🎓

🔍 Looking to enhance your problem-solving skills and dive deeper into algorithms and data structures using Python? Look no further!

📚 In this 30-session course:

- Refresh your Python basics

- Learn problem-solving strategies

- Master data structures like lists, stacks, and trees

- Solve real-world coding challenges on HackerRank

💡 Abdullah will guide you through each session, combining theory with hands-on practice to ensure you grasp every concept effectively.

👨‍💻 Whether you're an aspiring programmer or a seasoned coder seeking to sharpen your skills, this course is tailored for you!

#PythonProgramming #ProblemSolving #DataStructures #Algorithms

**Course Requirements:**

**1. Personal Computer and Internet Access:**

- Participants must have access to a personal computer or laptop with reliable internet connectivity throughout the duration of the course.

**2. Understanding of Basics of Computers:**

- Participants should have a basic understanding of computer operations, including familiarity with operating systems, file management, and basic software usage.

- Knowledge of fundamental computer concepts such as hardware, software, storage, and memory is beneficial but not mandatory.

**3. Basic Programming Knowledge (Recommended):**

- While prior knowledge of programming is not mandatory, having a basic understanding of programming concepts such as variables, control flow, and functions can be advantageous.

- Participants without prior programming experience will receive a "fresh-up" on the foundation of programming as part of the course curriculum.

**4. No Prior Knowledge of Python Required:**

- This course is designed for individuals with varying levels of programming experience, including beginners with no prior knowledge of Python.

- Participants will start with the basics of Python programming language, gradually progressing to more advanced topics in problem-solving, algorithms, and data structures.

**5. English Proficiency:**

- Since the course will be partially conducted in English, participants are required to have a sufficient understanding of the English language to follow instructions, read documentation, and engage in discussions.

- Basic proficiency in English is necessary to comprehend course materials, communicate with the instructor, and collaborate with peers effectively.

By meeting these requirements, participants will be well-equipped to engage with the course content and actively participate in learning activities, ensuring a productive and enriching educational experience in intermediate Python programming.

**Course Title: Intermediate Python Programming: Problem Solving with Algorithms and Data Structures**

Module 1: Introduction to Python and Basic Programming Concepts

Session 1: Introduction to Python

- Overview of Python programming language

- Installing Python and setting up the development environment

- Basic syntax, data types, and variables

Session 2: Control Flow and Functions

- Conditional statements (if, elif, else)

- Loops (for loops, while loops)

- Functions and parameter passing

- Scope of variables

Session 3: Lists and List Manipulation

- Introduction to lists and their properties

- Accessing and modifying list elements

- List methods and functions

Module 2: Problem Solving Fundamentals

Session 4: Introduction to Problem Solving

- Understanding problem-solving strategies

- Importance of algorithms and data structures

- Introduction to problem-solving techniques (brute force, divide and conquer, dynamic programming, etc.)

Session 5: Solving Problems with Python

- Analyzing and breaking down problems into smaller subproblems

- Writing Python code to solve simple algorithmic problems

- Practice problem-solving using HackerRank exercises

Session 6: Recursion

- Understanding recursion and recursive functions

- Recursive problem-solving techniques

- Examples and exercises on recursion

Module 3: Data Structures in Python

Session 7: Introduction to Data Structures

- Overview of data structures such as lists, tuples, sets, dictionaries

- Properties and applications of different data structures

Session 8: Stacks and Queues

- Understanding stack and queue data structures

- Implementation and applications of stacks and queues

- Solving problems using stacks and queues

Session 9: Linked Lists

- Understanding linked lists and their types (singly linked, doubly linked, circular linked)

- Implementation of linked lists in Python

- Manipulating linked lists and solving problems

Session 10: Trees and Binary Trees

- Introduction to trees and binary trees

- Properties and terminology associated with trees

- Implementing and traversing binary trees in Python

Each session will include a mix of theoretical concepts, practical coding exercises, and problem-solving challenges to reinforce learning and skills development. Additionally, regular assessments and quizzes will be conducted to track progress and understanding.

**Module 1: Introduction to Python and Basic Programming Concepts**

Session 1: Introduction to Python

- Overview of Python programming language

- Installing Python and setting up the development environment

- Introduction to Python interpreter and IDLE

- Writing and executing your first Python program

- Understanding Python's philosophy: readability and simplicity

Session 2: Basic Syntax, Data Types, and Variables

- Python syntax rules and conventions

- Data types: integers, floats, strings, booleans

- Variable naming rules and conventions

- Variable assignment and reassignment

- Type casting and conversions

- Using comments to document your code

Session 3: Control Flow and Functions

- Conditional statements: if, elif, else

- Comparison operators and logical operators

- Loops: for loops, while loops

- Iterating through sequences using loops

- Defining and calling functions in Python

- Passing arguments to functions

- Returning values from functions

- Scope of variables: local and global scope