

## Foundations of Programming: Python Course Summary

Course: IT FDN 110 B Au 23

**Objective:** This course has equipped me with a solid foundation in Python programming, covering both fundamental concepts and advanced techniques.

**Upon Completion, I have summarized what I learned as follows:**

- **Grasp Basic Programming Concepts:** Understanding the core principles of computer programming and their application in real-world scenarios. More specifically for me, this course provides support for the foundational understanding of important programming concepts such as constants, variables, methods, classes, etc.
- **Understand Software Design:** Learned about the software design process, including common trade-offs and decision-making strategies. During the last couple of modules, the lectures provided valuable insights into software design, especially the content covered during the module 8 lecture in which architectural diagrams were discussed as a potential first step in developing software. Furthermore, the additional discussion around test-driven development reaffirmed my understanding of the popular paradigm in software development.
- **Proficiency in Python:** Gained hands-on experience in writing non-trivial programs in Python, a modern, dynamically typed language.
- **Compare Programming Languages:** Capability around identifying key similarities and differences among various popular programming languages.

**This is a quick summary of the modules and my opinion on what was learned:**

1. **Python Basics:** Introduction to Python syntax and basic operations.
2. **Programming Basics:** Fundamental programming concepts, logic, and structures.
3. **Programming Tools and Techniques:** Exploration of essential programming tools and best practices.
4. **Collections of Data:** Understanding data structures like lists, tuples, and dictionaries.
5. **Advanced Collections and Error Handling:** Mastery of complex data structures and robust error handling techniques.
6. **Functions:** Deep dive into function definition, scope, and advanced features.
7. **Classes and Objects:** Object-oriented programming concepts, classes, and object manipulation.
8. **Advanced Programming:** Complex programming concepts, algorithms, and optimization.
9. **Special Topics:** Exploration of specialized topics, trends, and innovations in Python programming.
10. **Capstone Projects:** Practical application of learned skills in comprehensive projects.

**Application:** This course prepared me for real-world programming tasks, equipped me with both theoretical knowledge and practical skills. I will be able to develop, troubleshoot, and optimize Python programs for diverse applications with an added level of confidence.