Programming Foundations Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

This foundational training provides non-programmers, system administrators, and aspiring developers with a solid understanding of how computer systems work from a **programmer's perspective**.

Participants learn **core programming principles**, **data structures**, and **coding best practices** using Python for exercises.
The course is **language-neutral**, preparing learners to transition into other languages such as Java, C#, or C++.

Regulatory Note (BDDK)

While not a regulatory course on its own,**Programming Foundations** supports the **Banking Regulation and Supervision Agency (BDDK)** mandates — particularly **Article 20 (Secure Software Development)** and **Article 25 (Information Security Management)** — by instilling fundamental awareness of coding quality and secure development habits.

It is **recommended for financial institutions** as an introductory step toward secure software development compliance programs.

Who Should Attend

System administrators, support staff, trainee developers, and IT professionals looking to understand programming principles and collaborate effectively with developers.

Programming Foundations Training

Duration: **3 Day**

BILGINÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

By the end of this training, you will have gained knowledge and skills in the following areas:

- Understand computer and OS components.
- Work with data types, variables, and memory structures.
- Use operators and logical expressions effectively.
- Manage control flow, loops, and exceptions.
- Build modular programs with functions and libraries.
- Handle files and I/O operations safely.
- Write clean, readable, and maintainable code.

After completing this course, participants will:

- Have a clear understanding of software structure and execution,
- Be prepared to learn advanced languages like Java or C#,
- Possess the **foundational knowledge required for BDDK-compliant secure coding practices**.

Outline

Chapter 1: System Components

- Computer system components
- Central hardware components
- Moore's Law
- Software components
- Operating systems
- Processes
- Virtual memory
- Virtualisation
- Cloud Computing

Chapter 2: First Steps

- What is a program?
- What does a programmer do?
- Components of a program
- Programming languages
- Standards

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Applications and libraries
- Code used by more than one program
- What does a program do?
- Paper to program
- Creating a program
- A first program
- Running a program
- Error messages are your friends
- Giving names to data items
- Assignment statements
- Console vs. GUI
- Simple graphical messages
- Special characters

Chapter 3: Data

- Representing data
- Common numbering systems
- Numbers in programming languages
- Bits, bytes and words
- Conventions
- Fundamental types
- Getting it wrong
- Representing characters
- The problem with the Euro
- Representing integers
- Representing floating point
- E numbers
- Representing time
- Representing nothing

Chapter 4: Variables and Operators

- Variables and constants
- Objects
- Life of a variable - scope
- An alternative to scope
- Namespaces
- Choosing variable names
- Names you should not use
- Operations on data
- Choosing variable types
- Assignment and types
- Simple operations?
- Operator precedence
- Comments

Chapter 5: Containers

- Arrays and lists
- Sorting
- Other linear types - stacks, queues, deques
- Downside of linear structures

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Non-linear types and keys
- Containers for records - tree structures
- Associative arrays
- Compound types
- Class
- Object Orientation

Chapter 6: Flow Control

- Flow control
- Altering program flow
- Simple decision statements
- What is truth?
- Boolean operators
- Logical operators
- Using logical operators
- Loops
- Simple loop statements
- Array processing
- Language supplied iterators
- Interrupt handling
- Exception handling

Chapter 7: Program Structure

- Scope revisited
- Named blocks
- Calling a function
- Arguments and Parameters
- Passing arguments by copy
- Passing arguments by reference
- Returning results
- Function call syntax
- Recursion
- Entry points
- Modules and Libraries
- Why hide code and data? Encapsulation
- Asynchronous subroutines - Threads

Chapter 8: Input and Output

- What is a file?
- File systems
- Exchangeable file systems
- I/O Libraries and Layers
- File data types
- File names
- Opening a file
- Opening a file - checks
- Opening a file - modes
- Sequential access
- Random access
- Buffering

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Concurrency issues
- Locking strategies

Chapter 9: Building Programs

- Compilation
- Linking
- Loading and running
- Process attribute inheritance
- Portability
- Emulators
- Interpretation
- The third way: Byte-code
- Optimisation
- Debuggers

Chapter 10: Coding Style

- Virtues of a programmer
- Readability and style
- Naming conventions
- Error handling
- Programming for change
- The need for speed
- Programming for performance
- Constants - aren't
- Portability and flexibility
- Help!