Carbon Programming

Duration: 2 days Price: € 1499

Open Schedule

Carbon Programming

Audience Course Carbon Programming

The course Carbon Programming is intended for developers who want to learn about the possibilities of the Carbon programming language.

Prerequisites course Carbon Programming

To participate in this course, knowledge of a programming language such as C++, Rust, Java or C# is desirable.

Realization Training Carbon Programming

The course is a combination of theory and practice. The theory is discussed using slides and demos and is followed by practical exercises.

Course Certificate Carbon Programming

After successfully completing the course, attendants receive a certificate of participation in Carbon Programming.

Content Course Carbon Programming

In the course Carbon Programming participants learn to develop software in the next generation programming language Carbon. Carbon was launched by Google to possibly serve as a replacement for C++ in the future. Carbon is an open source project intended to solve the limitations of C++.

Intro Carbon Language

The course Carbon Programming starts with an introduction into the Carbon programming language covering its goals and design principles. Participants will gain a foundational understanding of Carbon as potential C++ successor and its associated modern tooling.

Variables and Data Types

Next attention is paid to variables and data types in Carbon, including primitive types, compound types, and type inference. Participants will learn how to declare, initialize, and manipulate variables of different data types in Carbon and will also learn about packages in Carbon.

Control Flow and Functions

Then control flow structures such as conditionals and loops are covered as well as functions and their parameters. Participants will learn how to write modular and reusable code using functions and control the flow of execution in Carbon programs.

Data Structures

Also data structures like arrays, lists, maps and sets in Carbon, along with their operations and usage are discussed. The handling of null values, pointers and generics are also on the program of the course Carbon programming.

Classes and Objects

Next object oriented programming concepts in Carbon including classes, objects, inheritance, virtual functions and polymorphism are treated. Participants will learn how to define and use classes to model real-world entities and create object instances in Carbon.

Memory Safety

Finally memory safety features in Carbon to prevent common programming errors such as memory leaks and buffer overflows, are explained. Carbon's memory management techniques and best practices for writing safe and secure code are covered.

Modules Course Carbon Programming

Module 1 : Intro Carbon Language	Module 2 : Variables and Data Types	Module 3 : Control Flow and Functions
Goal of Carbon	Comments	Control Flow
C++ Limitations	Declaration in Carbon	Conditionals
Carbon as Successor	var and let	Loops
Interoperability	Variables	while Loop
Modern Tooling	Data Types	for Loop
Package Manager	Numerics	match
Optimal Learning Curve	Strings	Functions
Build from Scratch	Type Inference	fn Keyword
Modular Code	Auto Keyword	Parameter Passing
Generics	Packages	Return Types
Module 4 : Data Structures	**Module 5 : Classes and Objects**	**Module 6 : Memory Safety**
Tuples	Objects in Carbon	Secure Memory Management
Multiple Coordinates	State and Behavior	Dangling Pointers
Pointers	Data Members	Buffer Overflows
Absence of Null Pointers	Member Functions	Spatial Memory Safety
Optionals	Class Methods	Array Boundary Checks
Arrays	Alias Declaration	Temporal Memory Safety
Indexed Access	Meaning of Self	Use After Free
Generics	Access Modifiers	Use After Return
Methods	Inheritance	Type Safety
Structs	Virtual Functions	Data Race Safety