

# Part IX. Functional Programming

In the functional programming model, functions are objects that, like other objects, can be passed as parameters to functions or stored in containers. There are numerous Boost libraries that support the functional programming model.

- Boost.Phoenix is the most extensive and, as of today, most important of these libraries. It replaces the library Boost.Lambda, which is introduced briefly, but only for completeness.
- Boost.Function provides a class that makes it easy to define a function pointer without using the syntax that originated with the C programming language.
- Boost.Bind is an adapter that lets you pass functions as parameters to other functions even if the actual signature is different from the expected signature.
- Boost.Ref can be used to pass a reference to an object, even if a function passes the parameter by copy.
- Boost.Lambda could be called a predecessor of Boost.Phoenix. It is a rather old library and allowed using lambda functions many years before they were added with C++11 to the programming language.

## Table of Contents

[39. Boost.Phoenix](#)

[40. Boost.Function](#)

[41. Boost.Bind](#)

[42. Boost.Ref](#)

[43. Boost.Lambda](#)