|  |
| --- |
| [Type the company name] |
| The Codex Language Specification |
| First Edition |
|  |
| **Colin Hill** |
| **2/22/2010** |

|  |
| --- |
| The Codex programming language is open source software and is licensed by the GNU Public License Version 3. |

Contents

[1 Introduction 3](#_Toc254634236)

[1.1 Example Programs 3](#_Toc254634237)

# 1 Introduction

The Codex programming language is a multi-paradigm, general purpose programming language. It has aspects of functional, object-oriented, and imperative concepts. Codex provides classes, inheritance, polymorphism, generics, garbage collection, function currying, and primitive data types such as integers, floating-point numbers, characters, lists, and arrays. Codex was conceived as a hobby and remains as such.

One of the main goals of Codex is to provide a clean and consistent concrete syntax. Codex provides syntactic features such as type-inference, function-currying, and naming convention enforcement to promote clear, concise, and consistent programs. The concrete syntax of Codex is designed as simply as possible to eliminate unneeded varying ways of defining the same thing. There are as few ways to write something as possible.

Codex is strongly and statically typed; types are statically determined at compile-time, whether they are defined explicitly in the code or inferred by the compiler.

## Example Programs

A simple Codex program consists of the main function, which takes the list of arguments and returns an integer value.

main(args: [String]) -> Int {

printLine(“Hello World”)

return 0

}

This simple program can be saved into a Codex source file “HelloWorld.cxs” and can then be compiled and run like so:

codexc HelloWorld.cxs

codex HelloWorld.cxb

When this program is run it will output “Hello World” to standard output.