article Sjors van Gelderen Exploring advanced programming concepts

amsmath [utf8]inputenc graphicx listings

document

Introduction In this document you will find a description of the material I have studied during my graduation phase. The primary focus of this project was to gain greater understanding of data structures, algorithms and complexity analysis. Secondly, I was interested in gaining greater proficiency in the use of the Python 3 and C# programming languages. Lastly, elementary asynchronous programming concepts were explored.

Languages Python 3 Advantages of this language include its concise syntax and its portability. Because Python 3 is a high-level language, it becomes unnecessary to worry about manual memory management. This is hugely advantageous during the initial exploration of an algorithm, as the programmer can focus exclusively on the workings of the algorithm itself.

A major downside of the language is the absence of a strict compiler. Python programs frequently crash during run-time, as problems are not detected at compile-time.

C# With Microsoft's recent decision to join the Linux Foundation, the .NET platform is becoming ever more attractive. The .NET platform is designed in such a way that its associated programming languages can directly interact with one another. This brings to the programmer the possibility to combine multiple programming languages in a single project with great ease. Since languages are designed with different problem domains in mind, this offers greater expressive power to the programmer.

C#, being perhaps the most popular .NET language, is an excellent choice to begin harnassing just this power. Since C# shares many similarities with the C/C++ programming languages, from which it derives its name, this language will be familiar to programmers who have experience with the aforementioned languages.

F# Being a more recent addition to the .NET family of programming languages, F# has not quite gained the popularity of C#. However, F#'s conception has been a step along the way to many of the modern functional programming features incorporated in C#. Regardless, F# is an excellent and powerful multi-paradigm (though mostly functional) programming language on its own.

Because of its functional nature, the F# programming language enables the programmer to tackle problems using recursion, higher order functions, partial application / currying, computation expressions (special syntactic sugar for monads) and more.

C Virtually any programmer will at some point in their career encounter this time-tested, fast and portable programming language. Because this low-level language doesn't use a garbage collector, the programmer must exercise great caution with the manual allocation and deallocation of memory. Many security problems that affect us today are a direct result of this fact.

Rust Developed by Mozilla, Rust aims to be a modern solution for asynchronous programming. With default immutable variables, borrowing and lifetimes, the compiler makes it very difficult indeed to write a program that contains run-time errors relating to incorrect memory access. A prime example of a project that suits Rust very well, is the Servo browser.

Chicken Scheme The LISP family of programming languages has two major dialects; these being Common LISP and Scheme. Chicken Scheme is a modern implementation of the Scheme dialect. It has a very minimalistic syntax, revolving around the use of parentheses and prefix notation.

Advanced C# features Properties This concept allows the programmer to specify how data inside a class may be accessed.

Interfaces Interfaces are contracts that guarantee a class implements certain methods.

Events Event-driven programming is made possible in C# through the use of

Extension methods In order to extend the functionality offered by base types, a programmer may use extension methods.

Higher order functions Higher order functions are part of what makes functional programming so powerful. In C#, delegates and lambdas offer the features required for higher order functions.

Iter Filter Map Fold LINQ

Anonymous types Using this feature, programmers may easily specify new types with a comfortable syntax.

lstlisting[Language=Python] var $some_type = SomeProperty = 5$;