# The Crystal Programming Language

Mehrshad Kavousi

University of Siegen, Germany

## Abstract

In this seminar paper, we provide a thorough introduction to the Crystal programming language; starting with its history and the motivations behind its emergence, to design principles, unique features, and well-known projects that utilize this language. At the end, we discuss our Crystal implementation of the **grep** utility, demonstrating the distinct Crystal features that helped us with writing this program.
**Keywords:** keyword 1, keyword 2, keyword 3, keyword 4, keyword 5, keyword 6

## 1 Introduction

- Brief overview of Crystal as a modern programming language

## 2 History and Development

- Founding, Versions and Motivations

# 3 Governance and Community

## 3.1 Core team

## 3.2 RFC

## 3.3 Sponsorship

# 4 Crystal Characteristics

## 4.1 Global Type Inference

## 4.2 Advanced Type System(Unions, NIL safety, etc.)

## 4.3 Crystal Compiler and LLVM

## 4.4 Garbage Collector (BDW)

# 5 Crystal Design Philosophy

## 5.1 Macro System and Compile-Time Metaprogramming

## 5.2 CSP-inspired concurrency model

## 5.3 Multi-threading

# 6 Infrastructure, Tooling, and Ecosystem

## 6.1 Package Management: Shards

## 6.2 The Build System and Static Linking

## 6.3 Documentation and Formatting

## 6.4 IDE Support

# 7 Current Usage

# 8 Security

# 9 Platform Support

# 10 Future Outlook and Roadmap

# 11 Implementation of grep utility in Crystal

## 11.1 Introduction

- An introduction of the grep utility, challenges in implementing, and an overview of the code.

### 11.2 Unique Crystal features used

- Which features unique to Crystal proved useful in writing the program.

### 11.3 Code Optimization

- What refinements and prudence employed in the code in order to improve time/space efficiency.
- Detailed empirical analysis of the implementation (Profiling)

## 12 Conclusion

## 13 Bibliography