

# Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
1.1	Programming Languages . . . . .	2
1.2	Compiler . . . . .	2
1.3	Programming Paradigms . . . . .	2
1.4	Code Quality and Design . . . . .	2
1.5	Operating System Concepts . . . . .	3
1.6	Algorithms . . . . .	3
1.7	Data Structure . . . . .	3
1.8	Distributed Systems . . . . .	3
1.9	Machine Learning . . . . .	3
<b>2</b>	<b>Compiler</b>	<b>3</b>
2.1	Parsers . . . . .	3
2.2	Lex . . . . .	3
2.3	Front-End . . . . .	3
2.4	Back-End . . . . .	3
2.5	IR . . . . .	3
2.6	Optimization . . . . .	3
<b>3</b>	<b>C++</b>	<b>3</b>
3.1	initilization . . . . .	3
3.2	value and copy semantics . . . . .	3
3.3	modules . . . . .	3
3.4	exceptions . . . . .	3
3.5	Idiom . . . . .	3

# **1 Overview**

abcdef

## **1.1 Programming Languages**

## **1.2 Compiler**

## **1.3 Programming Paradigms**

## **1.4 Code Quality and Design**

1. designe patterns
2. aspect/feature oriented programming
3. test driven development

**1.5**   **Operating System Concepts**

**1.6**   **Algorithms**

**1.7**   **Data Structure**

**1.8**   **Distributed Systems**

**1.9**   **Machine Learning**

**2**   **Compiler**

**2.1**   **Parsers**

**2.2**   **Lex**

**2.3**   **Front-End**

**2.4**   **Back-End**

**2.5**   **IR**

**2.6**   **Optimization**

**3**   **C++**

**3.1**   **Initilization**

**3.2**   **Value and copy semantics**

**3.3**   **Modules**

**3.4**   **Exceptions**

**3.5**   **Idiom**

**RAII**   **Resource ...**

**SFINAE**   **Resource ...**