

# 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 . . . . .	2
1.6	Algorithms . . . . .	2
1.7	Data Structure . . . . .	2
1.8	Distributed Systems . . . . .	2
1.9	Machine Learning . . . . .	2
<b>2</b>	<b>C++</b>	<b>3</b>
2.1	Initialization . . . . .	3
2.2	Value and copy semantics . . . . .	3
2.3	Modules . . . . .	3
2.4	Exceptions . . . . .	3
2.5	Idiom . . . . .	3
<b>3</b>	<b>Compiler</b>	<b>4</b>
3.1	Parsers . . . . .	4
3.2	Lex . . . . .	4
3.3	Front-End . . . . .	4
3.4	Back-End . . . . .	4
3.5	IR . . . . .	4
3.6	Optimization . . . . .	4

# **1 Overview**

## **1.1 Programming Languages**

## **1.2 Compiler**

## **1.3 Programming Paradigms**

## **1.4 Code Quality and Design**

1. design 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 C++**

### **2.1 Initilization**

### **2.2 Value and copy semantics**

### **2.3 Modules**

### **2.4 Exceptions**

### **2.5 Idiom**

**RAII** Resource ...

**SFINAE** Resource ...

## **3 Compiler**

### **3.1 Parsers**

### **3.2 Lex**

### **3.3 Front-End**

### **3.4 Back-End**

### **3.5 IR**

### **3.6 Optimization**