C++ and objects

Adam Boult (www.bou.lt)

February 10, 2023

Contents

Preface

This is a live document, and is full of gaps, mistakes, typos etc.

Part I

Basic changes from C

Chapter 1

Default function parameters

1.1 Introduction

1.1.1 Introduction

Chapter 2

Increments and decrements

2.1 Introduction

2.1.1 Introduction

Chapter 3

rvalues in C++

3.1 Introduction

3.1.1 Introduction

Chapter 4

auto

4.1 Introduction

4.1.1 Introduction

Chapter 5

Reference data types in C++

5.1 Introduction

5.1.1 Introduction

Chapter 6

Control flow

6.1 Introduction

6.1.1 Introduction

Chapter 7

Exception handling

7.1 Introduction

7.1.1 Introduction

Part II

Structs in C++

Chapter 8

Adding methods to structs in C++

8.1 Introduction

8.1.1 Introduction

Chapter 9

Struct inheritance in C++

9.1 Introduction

9.1.1 Introduction

```
struct point_2d {
   int x;
   int y;
};

struct point_3d: point_2d {
    int z;
};

point_3d my_point;
my_point.x = 1;
my_point.y = 2;
my_point.z = 3;
```

9.1.2 Multiple inheritance

```
struct point_2d {
   int x;
   int y;
};

struct colour {
  char red;
  char green;
```

```
  char blue;
};

struct point_3d_colour: point_2d, colour {
  int z;
};
```

Chapter 10

Static variables in structs in C++

10.1 Introduction

10.1.1 Introduction

Part III

Objects and classes

Chapter 11

Objects

11.1 Introduction

11.1.1 Keys and values

11.1.2 Classes

11.1.3 Integer caching

If we set $x = 2$ we can either create 2 in memory, or simply point x to 2, which is already in memory

That means if we do $x = 2$ $y = 2$ they have the same pointer.

Can also cache some other common data values, eg empty lists.

Makes sense if pointer is smaller in memory than value.

11.2 Representing objects

11.2.1 Representing a single object

11.2.2 Null in objects

11.2.3 Representing a class with a multiple array (ie 2d)

11.2.4 Representing a class with a single array (ie 1d)

11.3 Functions with objects

11.3.1 Creating new objects

11.3.2 Getting values by field

11.3.3 Adding fields

11.3.4 Changing values in fields

11.4 Hierarchies of objects

11.4.1 Inheritance

Chapter 12

Object-Oriented Programming

12.1 Introduction

12.1.1 Introduction

in objects, OOP. essentially, all variable types are objects. inc integers, floats, lists etc

Part IV

Generic programming

Chapter 13

Generic functions

13.1 Introduction

13.1.1 Introduction

Chapter 14

Generic classes

14.1 Introduction

14.1.1 Introduction

Chapter 15

Casting in C++

15.1 Introduction

15.1.1 Introduction

Part V

Compiling C++

Chapter 16

Object-Oriented Programming

16.1 Introduction

16.1.1 Introduction

Chapter 17

Object-Oriented Programming

17.1 Introduction

17.1.1 Introduction

Part VI

C++ libraries

Chapter 18

Packages and namespaces

18.1 Introduction

18.1.1 Introduction

Chapter 19

C standard library in C++

19.1 Introduction

19.1.1 Introduction

Chapter 20

C++ Standard Library

20.1 Introduction

20.1.1 Introduction

Part VII

Parallel programming in C++