ETH zürich

Exercise Session W14

Computer Science (CSE & CBB & Statistics) – AS 23

Overview

Today's Agenda

Follow-up
Objectives
Smart Pointers
Old Exams
Riddle
Q&A
Outro

`n.ethz.ch/~agavranovic`

1. Follow-up

Follow-up from previous exercise sessions

Follow-up from previous exercise sessions

- The silly mistakes from last time (regarding the dynamic allocation of arrays) are fixed and can be found on the handout on my webpage

2. Objectives

Objectives

□ Know where to find old exams (to practice with)

□ Clarify remaining questions

3. Smart Pointers

What?

What?

Smart Pointers are a (memory) safer alternative to ("raw") pointers

How?

```
#import <memory>            // to enable smart pointers
```

How?

```
#import <memory>            // to enable smart pointers
```

Now instead of raw pointers

```
int* scptr = new int(5);
```

How?

```cpp
#import <memory>                    // to enable smart pointers
```

Now instead of raw pointers

```cpp
int* scptr = new int(5);
```

You can just use (e.g. a unique) smart pointer

```cpp
std::unique_ptr<int> scUptr = std::make_unique<int>(5);
```

How?

```
#import <memory>                // to enable smart pointers
```

Now instead of raw pointers

```
int* scptr = new int(5);
```

You can just use (e.g. a unique) smart pointer

```
std::unique_ptr<int> scUptr = std::make_unique<int>(5);
```

The rest of the usage is almost the same.

- Unique pointers don't allow copying!

What for?

What for?

- They manage the memory for you!
- Leads to fewer memory leaks and other memory-related issues

Practicing with old exams

Practicing with old exams

- There are tons of old exams that can be found on the course website
- They can be a good way to practice for the real thing

Riddle

Riddle

Riddle me this code example on **code** expert . . .

6. Q&A

Q&A

Anything unclear?

7. Outro

General Questions?

See you at the exam!

Cheers!

$$\underbrace{\text{std::cout} \ll \underline{a}} \ll b$$

$$\underbrace{\text{std::cout} \ll b}$$