Source	master

...

Target	81f4d854

Commits (19)

Examples (/di918039/cs1pr-portfolio/commit/875ad95924237a6ee297bcd7b1dc42bbcf9101be) · 875ad959
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed 3 weeks ago

Constructors destructors (/di918039/cs1pr-portfolio/commit/d15b0c59f7cf5e5f962751b82283f811b820686e) · d15b0c59
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed 3 weeks ago

Nai (/di918039/cs1pr-portfolio/commit/5d7647ba95ed9c91db55527d0d3ddab19dc75700) · 5d7647ba
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed 3 weeks ago

Nai (/di918039/cs1pr-portfolio/commit/5b257911cf5a58be148a343a82465d4174ba7849) · 5b257911
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed 3 weeks ago

Further codes (/di918039/cs1pr-portfolio/commit/349cd563fa9ef873da21e3e751de528257464789) · 349cd563
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed 3 weeks ago

Further examples (/di918039/cs1pr-portfolio/commit/96ebc50ca5ebee6bf907af03834048d39442276b) · 96ebc50c
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed 3 weeks ago

Nai (/di918039/cs1pr-portfolio/commit/486516579e0637c5d05f3aef838a4c868cc8c83a) · 48651657
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed 3 weeks ago

Virtual destructors. (/di918039/cs1pr-portfolio/commit/ebf9a368f76f4957498a8e49ed4548e3b1186a9a) · ebf9a368
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed 3 weeks ago

Stuff. (/di918039/cs1pr-portfolio/commit/a7f0dc5be115409d6a8c7ff6f92f99c38017ea0d) · a7f0dc5b
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed 3 weeks ago

Further (/di918039/cs1pr-portfolio/commit/ecd0b885a3078ff5e14ca46a6e593ba050ddd390) · ecd0b885
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed 3 weeks ago

Nai (/di918039/cs1pr-portfolio/commit/6d03b95c0fb17e223c20e593dc99321164d2fbce) · 6d03b95c
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed 3 weeks ago

Nai (/di918039/cs1pr-portfolio/commit/609a7e696101c4daab846d1af87f421f962518b4) · 609a7e69
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed a week ago

Examples for 8 (/di918039/cs1pr-portfolio/commit/f23706f4eaa29c1204e8978c099c7b4975b68364) · f23706f4
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed a week ago

Exception 1 (/di918039/cs1pr-portfolio/commit/3d912c3bc45c05b8b02f64e25c609ba18414d909) · 3d912c3b
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed a week ago

Nai (/di918039/cs1pr-portfolio/commit/320c3ef059540c89c9475673f7a89f5f87c5e3b7) · 320c3ef0
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed a week ago

Smart pointer code (/di918039/cs1pr-portfolio/commit/8e0f875b544d093042bf3176945cccb266e39007) · 8e0f875b
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed about 22 hours ago

Nai (/di918039/cs1pr-portfolio/commit/2a354523c2cdfd16cc6166899422deb1812a76f9) · 2a354523
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed about 21 hours ago

Examples for Lecture 10 (/di918039/cs1pr-portfolio/commit/b174a3397a6fa0ba29df623874cefa4596a5b6d5) · b174a339
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed about 19 hours ago

Examples (/di918039/cs1pr-portfolio/commit/a6d904ef0ea60da9c938dd4ad8d710b40c1476d5) · a6d904ef
Julian M. Kunkel (mailto:juliankunkel@googlemail.com) committed about 17 hours ago

Showing **106 changed files** with **2319 additions** and **0 deletions**

lecture-examples/spring/10/const-expr.cpp 0 → 100644

```
+ #include <iostream>
+
+ using namespace std;
+
+ constexpr int varX = 10;
+
+ constexpr int get_five() {
+   return varX/2;
+ }
+
+ int main(){
+   int some_value[get_five() + 7]; // Create an array of 12 integers
```

```
13 +   return 0;
14 + }
```

lecture-examples/spring/10/lambda-simple.cpp 0 → 100644

```
 1 + #include <string>
 2 + #include <iostream>
 3 + #include <algorithm>
 4 + using namespace std;
 5 +
 6 + //This example may only work with recent compilers.
 7 + //clang++-9 -stdlib=libc++ lambda-simple.cpp
 8 +
 9 + int main() {
10 +   std::function<string()> funct = []() { return "Hello"; };
11 +   cout << funct() << endl;
12 +
13 +   auto ifunct = []() { return 5; };
14 +   cout << ifunct() << endl;
15 +
16 +
17 +   int multiplier = 2;
18 +   auto ifunct2 = [&multiplier](int v) { return v * multiplier; };
19 +   cout << ifunct2(6) << endl;
20 + }
```

lecture-examples/spring/10/lambda.cpp 0 → 100644

```
 1 + #include <iostream>
 2 + #include <string>
 3 + #include <vector>
 4 +
 5 + using namespace std;
 6 +
 7 + class Pallet
 8 + {
 9 +     public:
10 +         Pallet();
11 +         Pallet(vector<string> Items) : items(Items) {}
12 +         int GetWeight() { return items.size(); }
13 +
14 +     protected:
15 +         vector<string> items;
16 + };
17 +
18 + int main()
19 + {
20 +     vector<Pallet> pallets = {
21 +         Pallet({ "Scorpions" }),
22 +         Pallet({ "Dogs", "Bones", "Biscuits", "Cats", "Food", "Toys" }),
23 +         Pallet({ "Computers", "Scientists", "Routers", "Monitors" }) };
24 +
25 +     sort(pallets.begin(), pallets.end(), [](auto& a, auto& b) {
26 +       return a.GetWeight() < b.GetWeight();
27 +     });
```

```
28 + }
```

lecture-examples/spring/10/macro.cpp 0 → 100644

```
1  + #include <string>
2  + #include <iostream>
3  +
4  + #define ITEMS \
5  + TYPE(Computers), \
6  + TYPE(Scientists), \
7  + TYPE(Routers), \
8  + TYPE(Monitors),
9  +
10 +
11 + #define TYPE(e) e
12 + enum class item {
13 + ITEMS
14 + };
15 + #undef TYPE
16 +
17 + #define TYPE(e) #e
18 + const std::string item_strings[] = {
19 + ITEMS
20 + };
21 + #undef TYPE
22 +
23 + int main(){
24 +   std::cout << item_strings[(int) item::Routers] << std::endl;
25 + }
```

lecture-examples/spring/10/meta-gcd.cpp 0 → 100644

```
1  + #include <iostream>
2  + #include <string>
3  + #include <algorithm>
4  + using namespace std;
5  +
6  + template< int a, int b > struct GCD {
7  +     static const int RESULT = GCD< b, a % b >::RESULT;
8  + };
9  +
10 + template< int a > struct GCD< a, 0 > {
11 +     static const int RESULT = a;
12 + };
13 +
14 +
15 +
16 + int main() {
17 +   cout << "GCD (25,50) == " << GCD<25, 50>::RESULT << endl;
18 +   cout << "GCD (12,64) == " << GCD<12, 64>::RESULT << endl;
19 +   cout << "GCD (a,b) == " << GCD<18, 8398176>::RESULT << endl;
20 +
21 +   return 0;
22 + }
```