Exercises week 2

Klaas Isaac Bijlsma
s2394480

David Vroom
s2309939

November 28, 2017

Exercise 11

Exercise 12

Study the way ***`delete[]`*** *works*

We used the following code,

maxfour/maxfour.h

```
1  #ifndef EX12_MAXFOUR_H
2  #define EX12_MAXFOUR_H
3
4  #include <iosfwd>
5
6  class Maxfour
7  {
8      static size_t s_nObj;
9
10     public:
11         Maxfour();
12         ~Maxfour();
13     private:
14 };
15
16 #endif
```

maxfour/maxfour.ih

```
1  #include "maxfour.h"
2  #include <iostream>
3
4  using namespace std;
```

maxfour/data.cc

```
1  #include "maxfour.ih"
2
3  size_t Maxfour::s_nObj = 0;
```

maxfour/destructor.cc

```
1  #include "maxfour.ih"
2
3  Maxfour::~Maxfour()
4  {
5      --s_nObj;
6
7      cout << "Number of objects decreased by one (total: "
8           << s_nObj
9           << ")\n";
10 }
```

main.ih

```
1  #include "maxfour/maxfour.h"
```

main.cc

```
1  #include "main.ih"
2
3  int main()
4  try
5  {
6      Maxfour *array = new Maxfour[10];
7
8      delete[] array;     // In case the array is succesfully constructed
9  }
10 catch (...)
11 {}
```

Explain why the solution is so simple

The solution is so simple because when an exception is thrown during the construction of an array of 10 Maxfour objects, stack unwinding will destroy the already allocated objects. No explicit call of the destructor is needed. Furthermore we do not need to keep track of the already allocated objects.

Exercise 13

Exercise 14

Exercise 15

Exercise 16

Exercise 17

Exercise 18

Exercise 19