



INNOPOLIS
UNIVERSITY

Lab 2

C++ Classes, but without OOP

Ahror Jabborov, Asem Abdelhady, Daniel Atonge,
Marko Pezer, Mohammad Shahin

Software Systems Analysis and Design
Spring Semester 2024

Agenda

- Recap
- Code Style
- Constructors and Destructors
- Static vs Dynamic Object Declaration
- Exercises

Lecture Recap

- When is the heap memory used? What about stack memory?
- What is memory leakage, and with which type of memory is it associated?
- What are the four main types of constructors in C++?
- Constructors are for initializing an object's member data, what are the destructors for?
- What is the difference between *constexpr* and *const*?

Code Style

- Use Qt's coding style for C++.
- On Clion: go to Settings -> Editor -> Code style -> C/C++ -> Set from... Choose Qt.
- Find “format code” shortcut based on your OS.
- Some links: [JetBrains format code](#), [Qt coding style](#)

Constructors

```
1  class C
2  {
3  public:
4      int a;
5      C()
6          : a{0}
7          {} // default
8      C(int i)
9          : a(i)
10         {} // conversion
11      C(C &other)
12          { this->a = other.a; } // copy
13      C(int i, int j)
14          { a = i + j; } // ...
15  };
```

Constructors and Destructors

```
1  class Test
2  {
3  public:
4      int x;
5      Test()
6          : x(0)
7      {
8          cout << "Constructor is called here" << endl;
9      }
10     ~Test()
11     {
12         cout << "Destructor is called" << endl;
13     }
14 };
```

Dynamic vs Static Object Declaration

```
1  #include <iostream>
2  using namespace std;
3  class Test {...}
4  void
5  foo()
6  {
7      Test staticTest; // static declaration
8  }
9  int
10 main()
11 {
12     cout << "main starts here" << endl;
13     foo();
14     cout << "main ends here" << endl;
15     return 0;
16 }
```

Dynamic vs Static Object Declaration

```
1  #include <iostream>
2  using namespace std;
3  class Test {...}
4  void
5  foo()
6  {
7      Test *dynamicTest1 = new Test(); // dynamic declaration
8      auto dynamicTest2 = new Test(); // another option
9  }
10 int
11 main()
12 {
13     cout << "main starts here" << endl;
14     foo();
15     cout << "main ends here" << endl;
16     return 0;
17 }
```


Task 1

Write a program that contains a class *Box*.

- Box should have the length, width, and height as member variables.
The variables should be of type unsigned int.
- Box should have three constructors: default, copy, conversion.
- Box should have the assignment operator.

Task 2

Add and implement the following member functions to the class *Box*:

```
unsigned getVolume(); // returns the volume of the box.
```

```
void scale(unsigned scaleValue); // multiply each side of the box  
with scaleValue.
```

```
bool isBigger(unknown other); // this box is larger than the other  
(you decide what the type should be)
```

```
bool isSmaller(unknown other); // this box is smaller than the other  
(you decide what the type should be)
```

Task 3

Add the following operators to the class *Box*: { *, == }

** // Box with each side of the original box multiplied with an unsigned integer (scale value).*

== // returns true if the sides of two boxes are the same (they may be shuffled), false otherwise

Task 4

Add another class *Cube*. Add an operator to convert an object of type *Cube* to object of type *Box*.



INNOPOLIS
UNIVERSITY

The end.

Ahror Jabborov, Asem Abdelhady, Daniel Atonge,
Marko Pezer, Mohammad Shahin

Software Systems Analysis and Design
Spring Semester 2024