

# TDDD38/726G82:

## Adv. Programming in C++

Course Introduction

Christoffer Holm

Department of Computer and information science

- 1 What is TDDD38?
- 2 About C++
- 3 How to use C++
- 4 Basic IO

# What is TDDD38?

## Assumptions/prerequisites

In this course I will assume that:

- You can program in a procedural language

# What is TDDD38?

## Assumptions/prerequisites

In this course I will assume that:

- You can program in a procedural language
- You are familiar with object-oriented programming

# What is TDDD38?

## Assumptions/prerequisites

In this course I will assume that:

- You can program in a procedural language
- You are familiar with object-oriented programming
- You are **motivated** to become a better programmer

# What is TDDD38?

## Assumptions/prerequisites

In this course I will assume that:

- You can program in a procedural language
- You are familiar with object-oriented programming
- You are **motivated** to become a better programmer
- You are interested in a career involving programming

# What is TDDD38?

## Assumptions/prerequisites

In this course I will assume that:

- You can program in a procedural language
- You are familiar with object-oriented programming
- You are **motivated** to become a better programmer
- You are interested in a career involving programming
- **Note:** I'm **NOT** assuming that you know C++

# What is TDDD38?

What is programming *about*?

- Computation
- Abstraction
- Communication



# What is TDDD38?

What is programming *about*?

If a programmer wrote a solution that no one understands,  
is that a good solution?

# What is TDDD38?

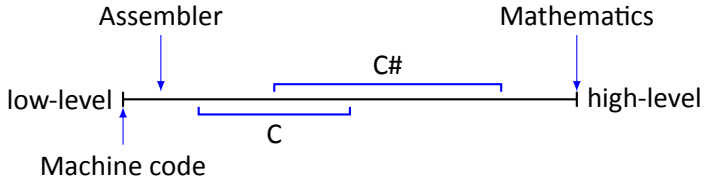
Why C++?

*“C makes it easy to shoot yourself in the foot; C++ makes it harder, but when you do it blows your whole leg off”*

– Bjarne Stroustrup

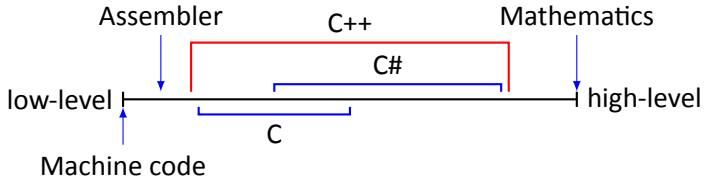
# What is TDDD38?

Why C++?



# What is TDDD38?

Why C++?



# What is TDDD38?

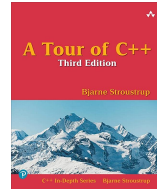
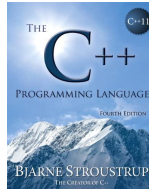
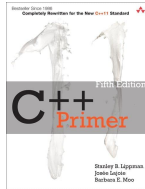
## Course information

- self-study
- lectures & seminars
- office hours
- Course web page:  
<http://www.ida.liu.se/~TDDD38/>
- E-mail: TDDD38@ida.liu.se
- Optional midterm test
- Examination

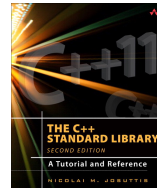
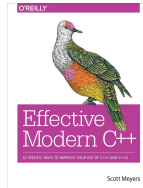
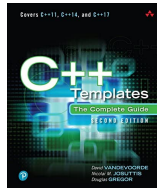
# What is TDDD38?

## Optional Literature

### General books:



### Specific books:



# What is TDDD38?

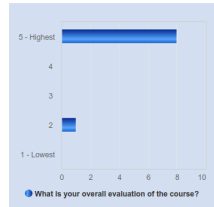
Literature

<https://en.cppreference.com/w/>

# What is TDDD38?

Evaluation from last time

- Students want mandatory labs
- **Answer:** I'm the only teacher so it's not possible due to time constraints

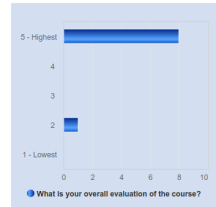




# What is TDDD38?

Evaluation from last time

- Students want the recorded lectures
- **Answer:** I never recorded the lectures during distance mode so *there are no recordings*



# What is TDDD38?

Changes from last term

- updated course goals
- Reduced time spent on basic object orientation
- A lot of updates to the slides and seminars/lectures

- 1 What is TDDD38?
- 2 **About C++**
- 3 How to use C++
- 4 Basic IO

# About C++

What is C++?

- general-purpose programming language

# About C++

What is C++?

- general-purpose programming language
- compiled language

# About C++

What is C++?

- general-purpose programming language
- compiled language
- based on C

# About C++

What is C++?

- general-purpose programming language
- compiled language
- based on C
- powerful tools for abstractions

# About C++

What is C++?

- general-purpose programming language
- compiled language
- based on C
- powerful tools for abstractions
- Designed by committee



# About C++

What is C++?

- general-purpose programming language
- compiled language
- based on C
- powerful tools for abstractions
- Designed by committee
- Has a standard ([ISO/IEC 14882:2020](#))

# What is C++?

## General rules

- C++ must be useful *now*
- Support different styles
- Forcing the programmer is bad

# What is C++?

## General rules

- All features must be affordable
- Usefulness > misuse prevention
- Composition of different parts

# What is C++?

## Technical rules

- No implicit violations of the type system
- User-defines types = built-in types
- Locality is good
- When in doubt, choose the easiest alternative

# What is C++?

Low-level rules

- Leave only assembler below C++
- Don't pay for what you don't use (zero-overhead)

# What is C++?

## Style

- No standardized style
- I will use a style, but you may use your own
- [Cpp Core Guidelines](#)
- This course will focus on C++17 and later
- Usage of C features will be penalized

- 1 What is TDDD38?
- 2 About C++
- 3 **How to use C++**
- 4 Basic IO

# How to use C++

## Tools required

- Operating System (duh!)



# How to use C++

## Tools required

- Operating System (duh!)
- Editor

# How to use C++

## Tools required

- Operating System (duh!)
- Editor
- Compiler

# How to use C++

## Additional tools

- Debugger

# How to use C++

## Additional tools

- Debugger
- Static analyzers

# How to use C++

## Additional tools

- Debugger
- Static analyzers
- Runtime analyzers

# How to use C++

## Compilers

Compilers	Linux	Mac	Windows
clang (clang++)	✓	✓	✓
GCC (g++)	✓	✓	✓ <sup>1</sup>
MSVC (cl.exe)	×	×	✓

---

<sup>1</sup>Ported as MinGW

# How to use C++

## Editors

- Emacs
- Vim
- Visual Studio Code
- etc.

# How to use C++

## IDEs

- CLion (All platforms)
- Eclipse (All platforms)
- Visual Studio (Windows)
- XCode (Mac)



# How to use C++

## How to compile

### **GCC:**

```
$ emacs file.cc           # write code
$ g++ -o myprogram file.cc # compile
$ ./myprogram             # run program
```

# How to use C++

## How to compile

### clang:

```
$ emacs file.cc           # write code  
$ clang++ -o myprogram file.cc # compile  
$ ./myprogram             # run program
```

# How to use C++

## How to compile

### MSVC:

```
$ emacs file.cc # write code  
$ cl.exe /Fe myprogram.exe file.cc # compile  
$ myprogram.exe # run program
```

# How to use C++

## Important terms

- Compile time (static)
- Runtime (dynamic)
- Implementation defined behaviour
- Undefined behaviour

- 1 What is TDDD38?
- 2 About C++
- 3 How to use C++
- 4 **Basic IO**

# Basic IO

**NOT** Hello World!

```
1 #include <iostream>
2 int main()
3 {
4     std::cout << "NOT Hello World!" << std::endl;
5     return 0;
6 }
```

## Basic IO

```
1  #include <iostream>
2  #include <string>
3  using std::cout;
4  using std::endl;
5  int main()
6  {
7      cout << "What is your name? ";
8
9      std::string name{};
10     std::cin >> name;
11
12     cout << "Your name is " << name << endl;
13 }
```

## Basic IO

```
1  #include <iostream>
2  #include <string>
3  using std::cout;
4  using std::endl;
5  int main()
6  {
7      int number{};
8      cout << "Enter a number: ";
9      std::cin >> number;
10     if (number >= 0)
11     {
12         cout << "Your number is positive!" << endl;
13     }
14 }
```



## Basic IO

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int counter{0}, sum{}, number{};
6      cout << "Enter your numbers: ";
7      while (counter < 5)
8      {
9          cin >> number;
10         sum += number;
11         ++counter;
12     }
13     cout << "The sum is: " << sum << endl;
14 }
```

## Basic IO

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int sum{}, number{};
6      cout << "Enter your numbers: ";
7      for (int i{0}; i < 5; ++i)
8      {
9          cin >> number;
10         sum += number;
11     }
12     cout << "The sum is: " << sum << endl;
13 }
```

# Basic IO

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int number{};
6      do
7      {
8          cout << "Enter number [0-10]: ";
9          cin >> number;
10     } while (number < 0 || number > 10);
11 }
```

# Basic IO

## Buffered input

```
1 int x;  
2 std::cout << "Enter integer: ";  
3 std::cin >> x;  
4 std::cout << "You entered: " << x << std::endl;  
5 std::cout << "Enter another integer: ";  
6 std::cin >> x;  
7 std::cout << "You entered: " << x << std::endl;
```

# Basic IO

## Buffered input

```
$ ./a.out  
Enter integer: 3  
You entered: 3  
Enter another integer: 5  
You entered: 5
```

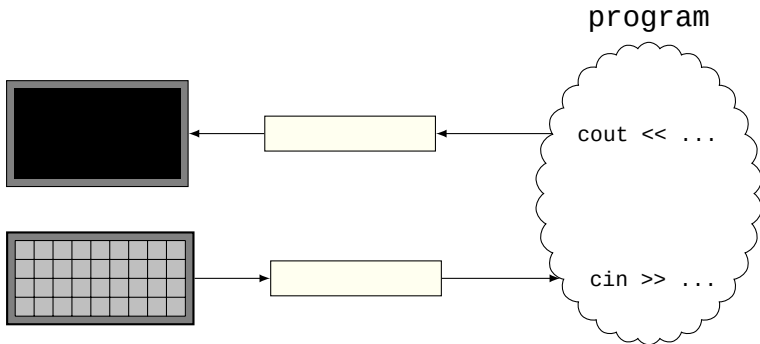
# Basic IO

## Buffered input

```
$ ./a.out  
Enter integer: 3 5  
You entered: 3  
Enter another integer: You entered: 5
```

# Basic IO

The complete picture



# Basic IO

## Buffered input

```
1  int x;  
2  std::cout << "Enter integer: ";  
3  std::cin >> x;  
4  std::cout << "You entered: " << x << std::endl;  
5  std::cin.ignore(1000, '\n');  
6  std::cout << "Enter another integer: ";  
7  std::cin >> x;  
8  std::cout << "You entered: " << x << std::endl;
```



# Basic IO

## Buffered input

```
$ ./a.out  
Enter integer: 3  
You entered: 3  
Enter another integer: 5  
You entered: 5
```

# Basic IO

## Buffered input

```
$ ./a.out  
Enter integer: 3 5  
You entered: 3  
Enter another integer: 7 8 9  
You entered: 7
```

[www.liu.se](http://www.liu.se)