Multimedia-Programmierung
Übung 3

Ludwig-Maximilians-Universität München

Sommersemester 2018

Today

A, B, C, C++

- procedural programming language with additional features
- C with Classes → object oriented extension of C
- Strong typing
- Provides facilities for low-level memory manipulation
- Influence for Java, C#, etc.

http://www.cs.tamu.edu/news/items?id=1797

Bjarne Stroustrup (born 30. December 1950 in Århus, Denmark) is a Danish computer scientist, most notable for the creation and development of the widely used C++ programming language.

Quelle: Wikipedia

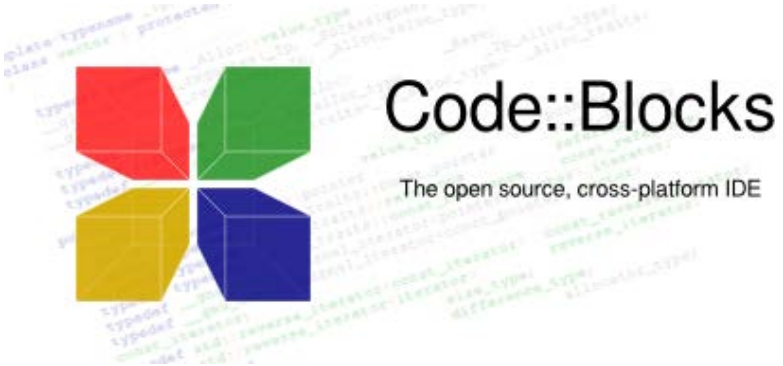
It’s just like Java, but...

Java	vs.	C++
Similar Syntax → if, else; for, while; etc.		
Easy to learn		More complex, but **high-performance**, efficient, flexible
Compilation to Virtual Machine		Native Compilation
Automatic **Garbage Collection**		DIY
One file per class		Separate declaration and implementation of classes → two files per class (**header files**)

Installation

Code::Blocks as IDE

and GNU Compiler



Linux/CIP Rechner → Code::Blocks

Windows → Code::Blocks + GNU or Virtual Studio

Mac → do not use Code::Blocks, instead use Xcode

http://www.codeblocks.org/

Download the 'codeblocks-16.01mingw-setup.exe' version

Get your Compiler Working

Code::Blocks needs to find the GNU Compiler.
If you get an error message saying that it can not find it follow these steps:

- Go to Settings -> Compiler
- In the Global Compiler Settings press the Reset defaults button and confirm the dialogs twice
- It should find the compiler automatically, if not check if you have downloaded the version from the last slide, that has the compiler included.

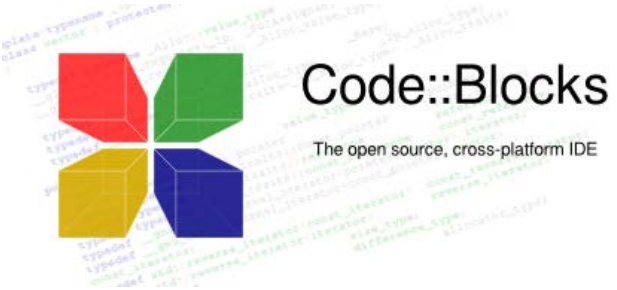
In the Global compiler Settings you can also choose the C++ standard. Newer versions of C++ have more features, that make programming easier and more efficient. For this tutorial choose the C++11 Iso C++ language standard.

Learning by Example: Hangman

For this tutorial we will program a small Hangman game in order to learn the basics of C++.

Hangman:
One player chooses a secret word. The other player has to guess letters that might be part of the word.
If the letter is included in the word its position gets revealed.
If the whole word is revealed the guessing player wins, but if the player guesses wrong 7 times the game is lost.

Step 1: Setup a new Project

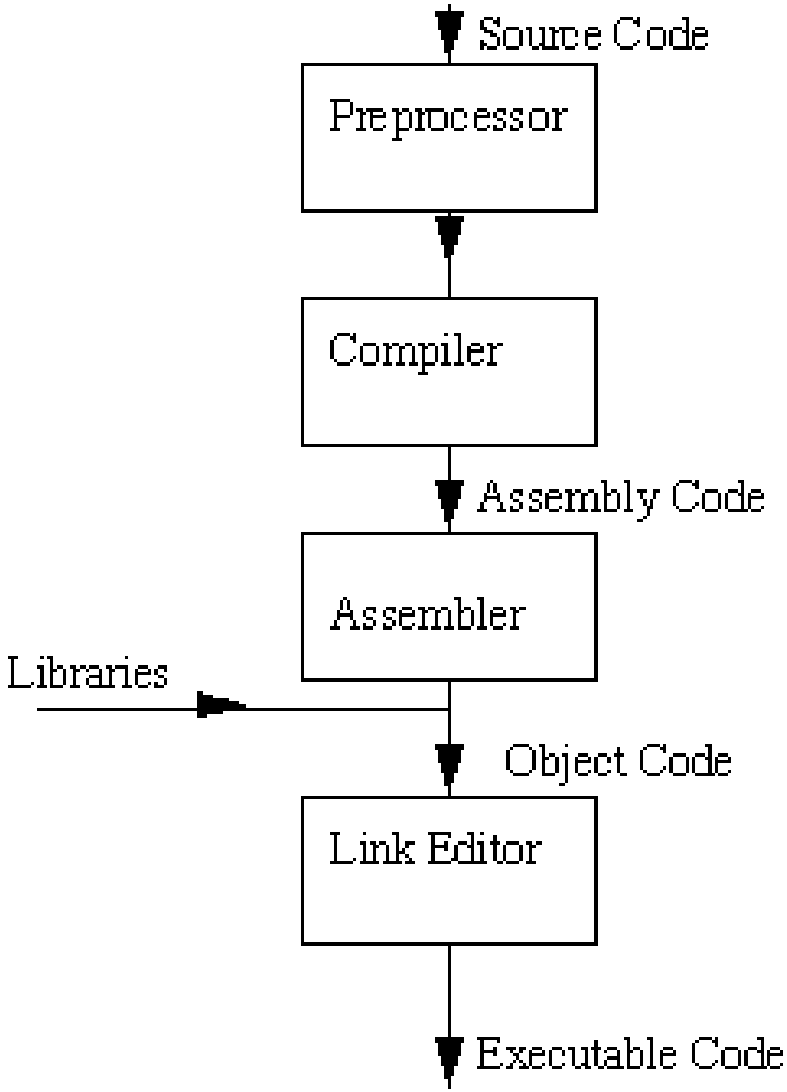


1. Open Code::Blocks
2. Go to File → New → Project
3. Choose Console Application
4. Click Next and choose C++ as a language
5. Give the Project a title, for example Hangman
6. Choose the GNU GCC Compiler
7. Finish
8. Build and Run
9. Add the files from the website to project

Knowledge 0: Compiling?!

Compiler Collection

- A collection of programs that are needed to create a program
- **Preprocessor** – Replaces Parts of Code in the Sourcefiles before the actual compile step
- **Compiler**: Translates Code fragments from C++ to machine code
- **Linker:** Combines compiled code fragments into executable program



Hello World C++ Style

```
#include <iostream>

int main()
{
   std::cout << “Hello World!” << std::endl;
   return 0;
}
```

- Like java the main method is called first, but no need for class
- Has to return something → 0 means no errors
- std is short for standard library → has a lot of useful functions and classes to make everything easier

Knowledge 1: Datatypes and Libraries

Primitive Datatypes:

- int
- Bool
- char [] aka C String
- char
- void

Important modules from the standard library:

<iostream>

- std::string
- std::cout and std::cin

<vector>

- std::vector<T>

<thread>

<math>

<exception>

<algorithms>

- for_each
- all_of

Step 2: The Puzzle Class

- Puzzle class will model actual Game

  Let‘s start by preparing the class:

  Task:
  Define attributes and methods in header file and later implement them in source file.

  Snippets 1-0, 1-1, 1-2:
  GameState

  Snippets 1-3:
  Define game setup and gameloop methods.

Knowledge 2: Classes and Header Files

- Header Files(.hpp) only contain definitions and declarations → the actual implementation in in the corresponding source file (.cpp)
- Other modules can then just #include the .hpp file and have access to all functions, classes, structs and enums of the module
- *“ Well, to be truthful, you don’t **have** to use header files… But you **should**!!!” “But why???? ☹”*
- **Because:** The code in x.cpp needs to be compiled only the first time or if it is changed; the rest of the time, the linker will link x’s code into the final executable without needing to recompile it. This makes rebuilding your program way faster.

Step 3: Instantiating and Calling Puzzle

Our main module (file with main-method) uses and controls the Puzzle class

Task:
Create an instance of the class and save it to memory

Snippet 2-0:
include Puzzle Class

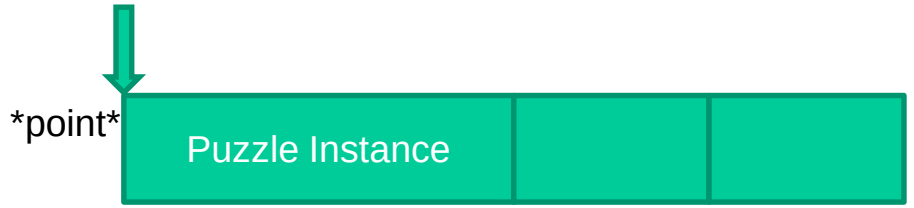
Snippet 2-1:
Puzzle Pointer, Constructor Call and game loop

Snippet 2-2:
Constructor implemenation

Knowledge 3: Fun with Pointers

- Constructor Call with new creates a new object somewhere in the computers memory.

- This location (address) can be assigned to ‘Puzzle* puzzle‘ pointer. Pointer points at beginning of object memory. (Stack vs. Heap)



&pointer: Gives us the address of the object it is pointing to

***pointer**: Gives us the value of the same.

puzzle->startGame(): with pointer we need → intstead of . in order to call a method.

Step 4: Prepare and Start the Game

Task:
Add attributes to Puzzle and implement the startGame method

Snippet 3-0
Add attributes

Snippet 3-1:
- *startGame implementation*
- *set the attributes to starting values*
- *secret word input*

Step 5: Game Logic

Task:
Implement the actual game logic

Snippet 4-0
The gameloop methods is implemented

But wait… WTF is happening???

Let's look at an example:

The player enters a puzzle word:

puzzle word = "hello"

The program now creates a list of boolean values of the same size as the string:

guessed = [false, false, false, false, false]

Now another player enters a letter to guess:

char input = 'l'

Now the program looks for the letter in the string and finds it twice, setting the corresponding boolean values to *true:*

guessed = [false, false, true, true, false]

WTF is happening??? (contin.)

If the program can’t find the letter it subtracts a try from the amount of tries the player has. If after subtracting the amount of tries is 0, the game is over.

Now the program looks through the boolean array. If the element is 'false' it prints an '_', if it is true it prints the letter from the puzzle word, that is at the current index.

Word = soap
guessed = [false, false, true, false]
Console: _ _ a _

Finally the programm looks through the guessed list. If all values are true the game is over and the guessing player has won.

Step 6: Console Graphics

Task:
Draw a hangman on the console

Snippet 5-0
Drawing

Want to know more?

- API: http://www.cplusplus.com/reference/
- The BOOK: https://upload.wikimedia.org/wikibooks/de/5/59/Cplusplus.pdf
- Tutorials: http://www.cplusplus.com/doc/tutorial/
- Recommended: http://www.learncpp.com/

Preview on Cocos:

- https://www.youtube.com/watch?v=qXqgSNUf9Cc&list=PLRtjMdoYXLf4od_bOKN3WjAPr7snPXzoe
- http://wizardfu.com/book/cocos2d-x/