CH-230-A

Programming in C and C++

C/C++

Lecture 1

Dr. Kinga Lipskoch

Fall 2020

Who am I?

- PhD in Computer Science at the Carl von Ossietzky University of Oldenburg
- University lecturer at the Computer Science Department
- Joined Jacobs University in January 2013
- Office: Research I, Room 94
- Telephone: +49 421 200-3148
- E-Mail: k.lipskoch@jacobs-university.de
- Office hours: Mondays 10:00 - 12:00 (online call using Teams)

Course Goals

- Learn the basic and some advanced aspects of procedural and object-oriented programming
- Learn the details of the C and C++ programming languages
- Write, run, test, debug programs using C and C++

Course Details

- Every week will consist of
  - 2 tutorials: Tuesdays, 8:15 - 11:00
  - 1 lecture: Thursdays, 11:15 - 12:30
- During each tutorial you will have to solve a programming assignment sheet (consisting of multiple exercises) related to the corresponding lecture
- Presence assignments need to be submitted during the tutorial time
- Other assignments can be submitted before next Monday at 23:00
- Help session will be offered by TAs before the deadline: Sundays, 19:00 - 21:00, online in Teams

Course Resources

- Homepage of course: `https://grader.eecs.jacobs-university.de/courses/ch_230_a/2020_2/`
  Slides, assignment sheets and practice sheets will be posted there
- Program assignments will be received during the tutorial
  Presence exercises have to be submitted during the tutorial
- Offline questions:
  Office hours on Mondays 10:00 - 12:00 (using Teams)
- Do not hesitate, and do not wait until you are left too much behind

Literature

Some example textbooks for C and C++ are:

- Brian Kernighan, Dennis Ritchie:
  The C Programming Language
- Steve Oualline:
  Practical C Programming
- Bruce Eckel:
  Thinking in C++: Introduction to Standard C++
- Bruce Eckel, Chuck Allison:
  Thinking in C++: Practical Programming
- Bjarne Stroustrup:
  The C++ Programming Language
- Michael Dawson:
  Beginning C++ Through Game Programming

Beyond this Programming Course

- Programming skills might be one of the key career parametres
- Programming itself is a vast and multi-faceted activity - mixture of right mental attitude, skills and experience
- This course is only there to get you started
- Internet has a plenty of resources and cookbook recipees - but good initial skills are necessary to profit from them
- Several online portals allow you to hone your skills by solving problems and collect "badges" or have "reputation" score
  - `https://leetcode.com`
  - `https://hackerrank.com`
  - `https://stackoverflow.com`
  - `https://www.spoj.com`
  - ...

Submission of Solutions

- Use Grader
  `https://grader.eecs.jacobs-university.de/`
  with your CampusNet credentials
- Submit `*.c` or `*.cpp` and `*.h` files depending on the problem (which language to use will be fixed for each problem)
- Pay attention to deadlines and submit before

Grader not Publicly Visible

- You can access Grader from campus without any additional connection or software
- To access Grader from outside of campus you need to use a VPN (Virtual Private Network) connection
- Instructions from the Jacobs IRC IT team on how to install a VPN client: `https://teamwork.jacobs-university.de/display/ircit/VPN+Access`

Grading

- Each problem will be graded (percentages)
- ug 23: 67% assignments grade, 33% final exam grade, both components have to be passed with at least 45%
- ug 22: to attend final exam you need to have $\geq$ 50% as average over all assignments, 100% final exam grade
- other ug: please write me an e-mail
- In the (written) final exam you will be asked to solve exercises similar to ones in the assignments
- The final exam will take place at the end of the semester scheduled by the Student Records Office

Grading Criteria for Assignments

- Assignments are reviewed by the TAs
- Not just the solution counts, but programming style and form
- TAs will follow guidelines according to document `https://grader.eecs.jacobs-university.de/courses/ch_230_a/2020_2/Grading-Criteria-C-C++.pdf`
- They will use rules to review your solutions
- Two types of assignments:
  - Automatically graded problems with testcases only - for example, 10 uploaded testcases and your solution passes only 7 then your grade will be 70%
  - Manually graded problems - with feedback from TAs and a grade between 0% and 100%

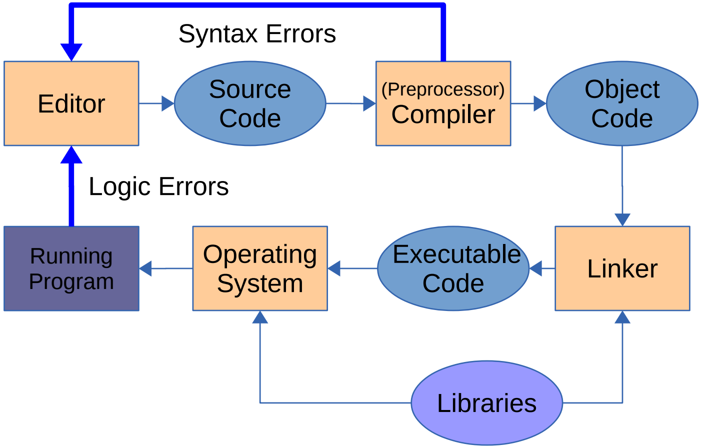
Excuse from Academic Obligations according to AP

- `https://www.jacobs-university.de/sites/default/files/bachelor_policies_v_3.4.pdf` (pages 14 - 15)
- Illness must be documented with a sick certificate
- Sick certificates need to verify the date and time of the in-person visit occasioned the confirmation that the student is unable to fulfill his/her academic obligation (either attend class/lab or take the examination)
- Sick certificates and documentation must be submitted to Registrar Services by the third calendar day from the beginning of illness/of the emergency
- These three days include the first day of the illness/of the emergency
- If the third calendar day is a Saturday, Sunday or a public holiday, the deadline is extended to the next working day.
- Predated or backdated sick certificates, i.e., when the visit to the physician takes place outside of the documented sickness period will be accepted provided that the visit to the physician precedes or follows the period of illness by no more than one working day
- Regardless of the reason for their absence, students must inform the IoR before the beginning of the examination or class/lab session that they will not be able to attend
- The day after the excuse ends, students must contact the IoR
- Failure to complete a module will lead to a continued incomplete of the module until the missing requirements are fulfilled or definitively failed

Prerequisite for Algorithms and Data Structure

- This course is a prerequisite for the second semester course "Algorithms and Data Structures" (ADS)
- If you fail or do not take the exam for this course you will not be allowed to proceed with the ADS course
- ADS is a prerequisite for almost all second year CS courses
- If you fail the first exam (December) you can take the so-called make-up exam (January)

Program Development Cycle



Integrated Development Environment (IDE)

- You can use the editor of your choice and compile from the terminal
- For C: `gcc -Wall -o executable program.c`
- For C++: `g++ -Wall -o executable program.cpp`
- If you do not know any of the above, you can use Code::Blocks or Visual Studio Code

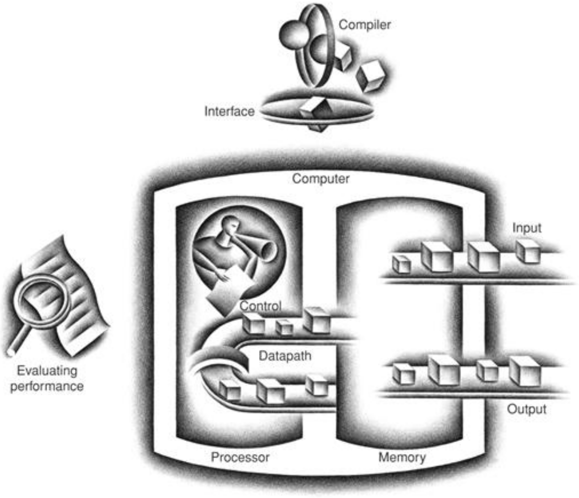
IDE Installation

- Alternative 1: Code::Blocks
  - Download and install Code::Blocks from: `http://codeblocks.org/downloads/26`
  - If you are a Windows user download (contains IDE + compiler) `codeblocks-20.03mingw-setup.exe`
- Alternative 2: Visual Studio Code
  - Download and install Visual Studio Code `https://code.visualstudio.com/download?wt.mc_id=DX_841432`
  - Download and install compiler Mingw-w64 `http://mingw-w64.org/doku.php/download/mingw-builds`
  - Assuming that you installed Mingw-w64 to this path: `C:\Mingw-w64`, Windows users have to add to the environment variable `PATH` in the following: `C:\Mingw-w64\mingw32\bin\`
- Alternative 3: Visual Studio Community 2019
  - Only for Windows users
  - Download and install `https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&rel=16`

Different Compilers Behave Differently

- Different compilers behave differently
- Even different versions of the same compiler may deliver different results in terms of the compilation process
- Make sure that your solution runs without warnings or errors on Grader
- If errors of warning appear, you can fix them and resubmit the solution
- The Grader server runs `gcc` and `g++` version `8.3.0`

Five Classic Components of the Computer

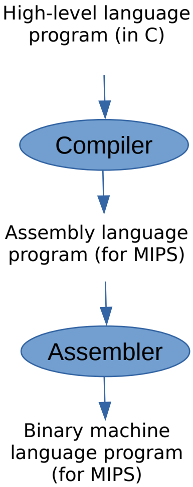


Higher Programming Languages

- Use symbolic names
- Loops, conditions

High-level language program (in C)

```
swap(int v[], int k) {
  int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```


High-level language program (in C)
Compiler
Assembly language program (for MIPS)
Assembler
Binary machine language program (for MIPS)

```
swap(int v[], int k) {
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

```
swap:
    muli   $2, $5, 4
    add    $2, $4, $2
    lw     $15, 0($2)
    lw     $16, 4($2)
    sw     $16, 0($2)
    sw     $15, 4($2)
    jr     $31
```

```
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Compiling and Running

- The following command performs compilation and linking
  ```
  $> gcc -o hello hello.c
  ```
- If no compilation errors, an executable called `hello` will be created
- If you do not specify o the executable will be called `a.out`
- To execute the program just type its name
  ```
  $> ./hello
  ```

About C

- Widely used general purpose language
- Advantages: small, efficient, portable, structured
- Disadvantages: not user-friendly
- C is an imperative language
- You will find many of its characteristics in other imperative languages, such as Pascal or Fortran, but also in scripting languages such as Perl, PHP, Python, etc.

Imperative Languages

- Computation is described in terms of:
  - State (variables)
  - Operations to change this state (assignments, loops, etc.)
- Imperative: first do this, than do that, ...
- There exists other approaches (functional programming, logic programming, object-oriented programming, etc.)

The First Program

- A true classic: Hello world
- Open editor → New file
- Type text below, then save as `hello.c`

```c
/* This is my first C program */
#include <stdio.h>

int main() {
  printf("Hello world\n");
  return 0;
}
```

The printf Library Function

- printf is a library function used to output data
- To use printf, the header file stdio.h has to be included
- stdio stands for Standard I/O
- stdio contains the specification of many general purpose functions for I/O
- printf is a very rich and powerful function
- Basic use: printing a sequence of characters
- The following line calls the printf function
  ```
  printf("Hello world\n");
  ```
- The sequence of characters is called string
- The sequence is surrounded by quotes

Basic Data Types of C

Data type	**C identifier**
Character	`char`
Integer number	`int`
Floating point number	`float`
Double precision number	`double`
No type	`void`

Moreover there exist some modifiers that can be applied to the basic data types

Formatting Specification (1)

- Specify the type of the data to be printed

```c
int a = 45;
printf("The value is %d\n", a);
```

- This will print the following:

```
The value is 45
```

- Each formatting specification must be matched by a parameter
- To specify wrong control strings is another common error in C programs

Formatting Specification (2)

Formatting specification starts with a % character

Conversion	**Meaning**
`%c`	Single character
`%d` or `%i`	Signed decimal integer
`%f`	Floating point (decimal notation)
`%e`	Floating point (exponential notation)
`%lf`	Double (decimal notation)
`%s`	String
`%%`	print the percent sign itself
`%p`	print address of pointer