Tutorial 13 - 04.02.2020

Group 06 - Moritz Makowski

Revision

Today's slideshow will only include the list of keywords I will talk about during this revision-tutorial.

Today's Agenda

- Software Engineering
- Variables
- Arrays
- Input/Output

- Loops
- Conditionals
- Functions

- Pointers
- Memory
- Structs, Enums, Unions

- Lists
- Stacks, Queues
- Trees
- Sorting

- Control Flog Graphs
- Bitwise Operators

Software Engineering

- What is programming?
- What is compiling?
- How do you compile a program?

Variables

- Types of variables
- What is scope?
- What is type casting?

Arrays

- What are arrays?
- How to access/manipulate an array
- Multidimensional arrays

Input/Output

- What are strings? How can you store them in C?
- Basic output function
- Basic input function

Loops

- What types of loops are there?
- When should you use which type?
- What does `break` and `continue` mean?

Conditionals

- What are boolean expressions?
- Which boolean operators do you know?

- Two common ways of conditionally executing code?
- Syntax of the two ways

Functions

- Why should we use functions? (As many reasons as you can think of)
- Basic executional flow when using functions
- Function syntax passing arrays to functions

Pointers

- What are pointers?
- What is referencing and dereferencing?
- Syntax of (de-)referencing

- Arrays and pointers?
- Passing by reference vs. passing by value

Memory

- Difference between static and dynamic memory allocation
- How to dynamically allocate memory?
- What has to be done with dynamically allocated memory after execution?

Structs, Enums, Unions

- Why should you use structs?
- Struct syntax

- Why should you use enums?
- Enum syntax

- Why should you use unions?
- Union syntax

Lists

- What types of lists are there?
- Benefits and drawbacks of each type?
- Basic implementation of each type

- *(Bonus)* How may you combine the two types we mainly talked about?

Stacks, Queues

- How does a stack work?
- Find an example in the real world and a use case in computer science

- How does a queue work?
- Find an example in the real world and a use case in computer science

Trees

- Difference between a regular tree and a binary tree
- Prime usecase for binary trees?
- Name the different parts of a tree
- What is an adjacency matrix?

- How to implement a tree in C?

- What is tree-traversal?
- What kinds of traversal do you know?

- What is binary search?

- Usecase: Syntax Trees

Sorting

- How does bubble sort work?
- How does merge sort work?

- Which of the above is more efficient?
- Types of efficiency
- How can we describe efficiency?

Control Flow Graphs

- Why should we use CFG

- How to draw a CFG from code
- How to generate code from a CFG

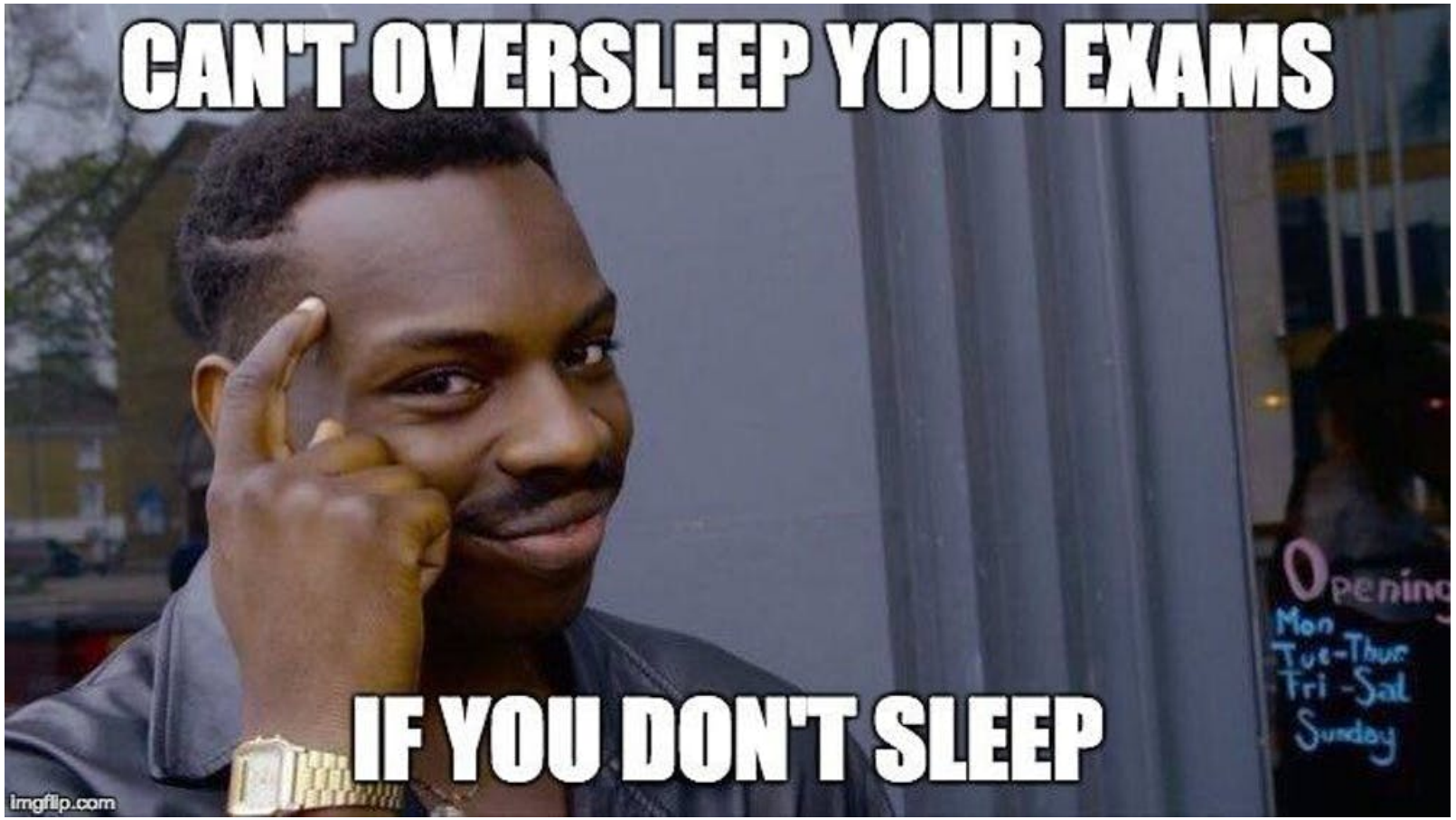
Bitwise Operators

```c
int a = 43;
int b = 3678;
int c;

c = a << 3; // left shift
c = a >> 2; // right shift

c = ~a; // bitwise NOT

c = a & b; // bitwise AND
c = a | b; // bitwise OR
c = a ^ b; // bitwise XOR
```

Have a look at `example_13_1_bitwise.c` *on GitHub.*

Good Luck for the Exam!

All **code examples** and **exercise solutions** (available right after my tutorial) on **GitHub**.
https://github.com/dostuffthatmatters/Engineering-Informatics-1-MSE-WS1920.


CAN'T OVERSLEEP YOUR EXAMS
IF YOU DON'T SLEEP
Opening
Mon
Tue-Thur
Fri -Sat
Sunday
imgflip.com