Einführung in C – Introduction to C

9. Real-life C projects

Prof. Dr. Eckhard Kruse

DHBW Mannheim

From hello-world to real-life projects

Quick overview

data types and operators

functions and program structure

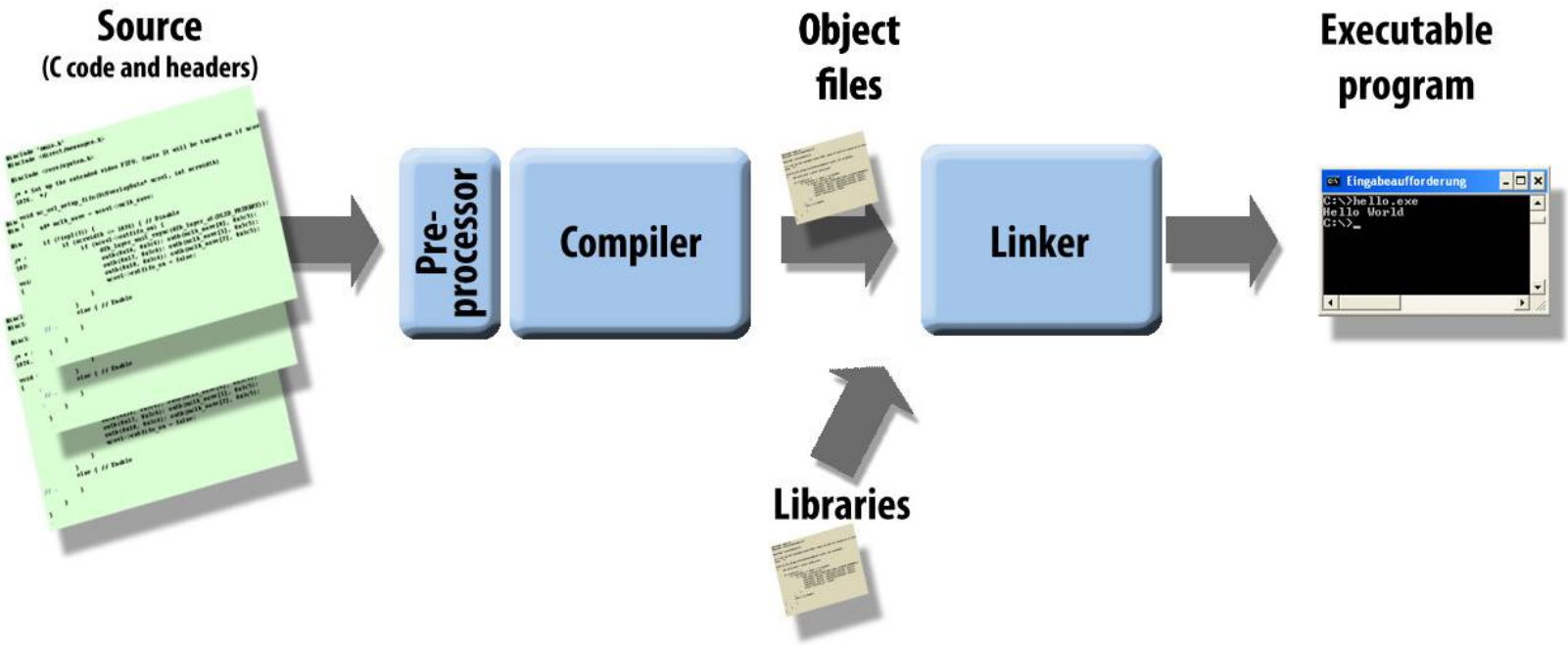
pointers and memory management

control flow

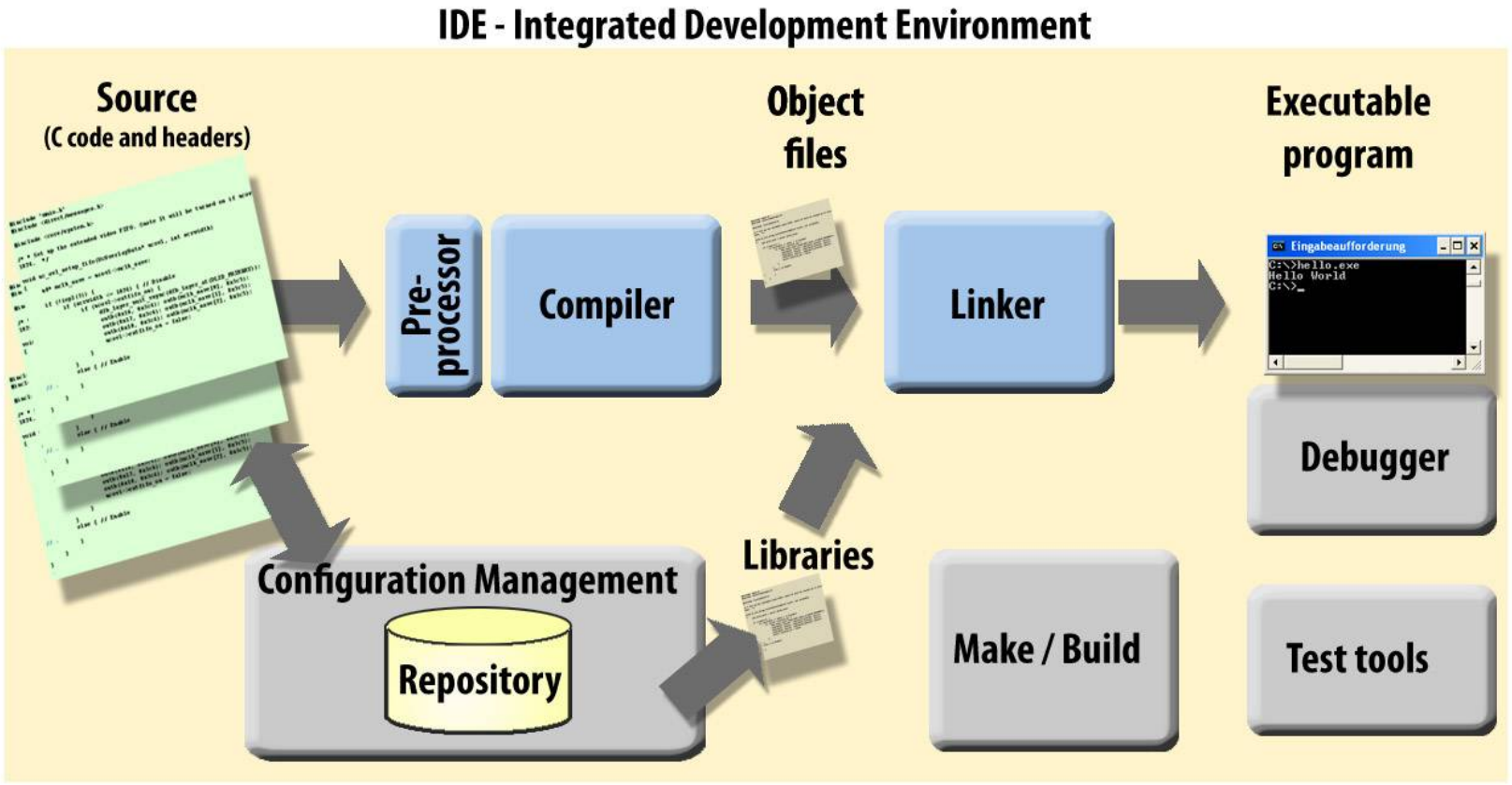
libraries

real-life C projects

Simple workflow



Integrated development environment



Make / Build

Real-life projects can consist of hundreds or thousands of source code files and numerous additional libraries. Everythings needs to compiled and linked in a specific order, taking into account configuration settings (e.g. to build different versions for debugging and release).

To do this efficiently, tools/scripts for automatic build are used.

Simple makefile example

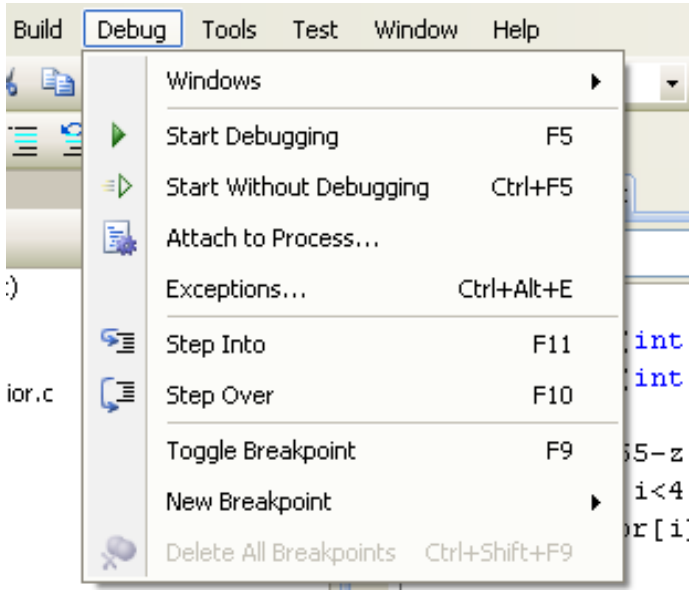
```
CFLAGS = -g -Wall
CC = gcc
LIBS =  -lm
INCLUDES =
OBJS = main.o ui.o logic.o math.o
SRCS = main.c ui.c logic.c math.c prog1.c prog2.c
HDRS =  ui.h logic.h math.h

all: prog1 prog2

prog1: prog1.o ${OBJS}    ${CC} ${CFLAGS} ${INCLUDES} -o $@ prog1.o $
{OBJS} ${LIBS}
prog2: prog2.o ${OBJS}    ${CC} ${CFLAGS} -o $@ prog2.o ${OBJS} ${LIBS}

.c.o: ${CC} ${CFLAGS} ${INCLUDES} -c $<

depend:    makedepend ${SRCS}

clean:rm *.o core *~
```

Debugger

Most development environments provide a **debugger** that allows supervised execution of the program, including step-by-step execution of the program code and display of actual variable values. This is a crucial tool for finding bugs in the program.

Have a look at the debugger in your development environment:

1) Build a debug version of your program.
2) Set breakpoints.
3) Run the program (to the breakpoints).
4) Use step-by-step execution
5) Have a look at the values of your variables
6) Have a look at the call stack



C Style Guides

A good coding style improves readability and can help reduce errors.
Conventions might regard e.g.:

- Use of Spaces, identation and blank lines
- Statements, braces (e.g. one statement per line, braces stand alone)
- Comments (content, where to be placed, formatting)
- Naming (variables, functions, upper/lower case, underscore)
- Function + program structure
- File organization
- Information hiding and variable scopes
- Use of libraries, compliance to standards (e.g. C99)
- ...

Example …

Get to know real-life programming

Exercise

Get to know real-life projects!

The Web is full of them, see e.g. github.

How to recognize a good programmer?

http://www.inter-sections.net/2007/11/13/how-to-recognise-a-good-programmer/
How do you recognise good programmers if you're a business guy?
(Assumption: The CV does not help much)

Positive Indicators:

- Passionate about technology
- Programs as a hobby
- Will talk your ear off on a technical subject if encouraged
- Significant (and often numerous) personal side-projects over the years
- Learns new technologies on his/her own
- Opinionated about which technologies are better for various usages
- Very uncomfortable about the idea of working with a technology he doesn't believe to be "right"
- Clearly smart, can have great conversations on a variety of topics
- Started programming long before university/work
- Has some hidden "icebergs", large personal projects under the CV radar
- Knowledge of a large variety of unrelated technologies (may not be on CV)

How to recognize a good programmer?

http://www.inter-sections.net/2007/11/13/how-to-recognise-a-good-programmer/
How do you recognise good programmers if you're a business guy?
(Assumption: The CV does not help much)

Negative Indicators:

- Programming is a day job
- Don't really want to "talk shop", even when encouraged to
- Learns new technologies [only] in company-sponsored courses
- Happy to work with whatever technology you've picked, "all technologies are good"
- Doesn't seem too smart
- Started programming at university
- All programming experience is on the CV
- Focused mainly on one or two technology stacks (e.g. everything to do with developing a java application), with no experience outside of it

Being a good programmer?

Exercise

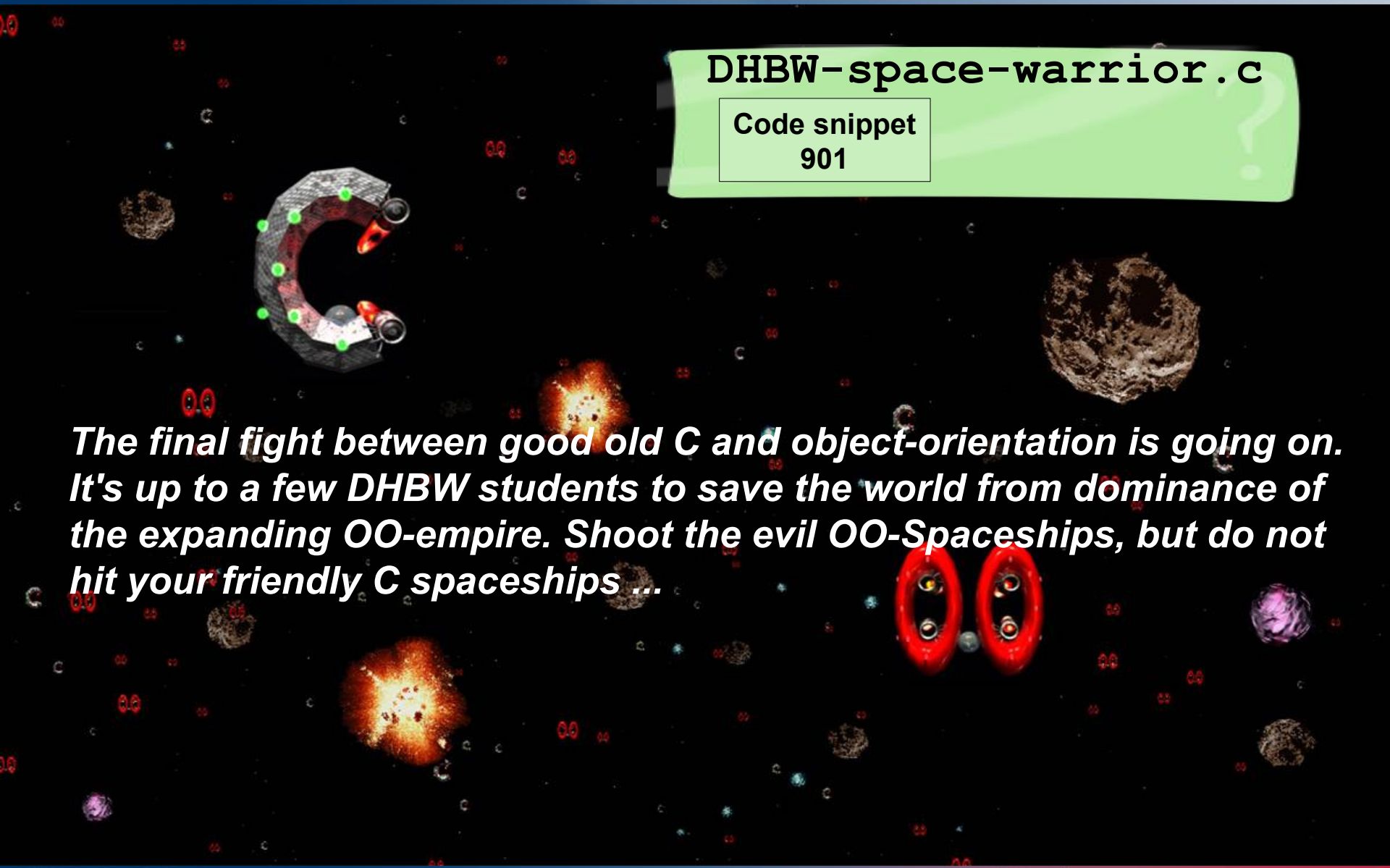
Do I want to be a good programmer?

You don't become a good progammer by attending lectures but by doing lots of programming (which means you should love it...)

Reflect upon your personal interests:

1) To what degree do you fulfill the checklist on the previous slides?
2) Do you agree with these indicators, or would you propose other?
3) Discuss: How important are programming skills for an IT career? (Distinguish between potentially different IT careers)
4) Any conclusions about how/whether you want to strengthen your programming skills in the future?
5) Discuss with your neighbour.

DHBW Space warrior



DHBW-space-warrior.c

Code snippet
901

The final fight between good old C and object-orientation is going on. It's up to a few DHBW students to save the world from dominance of the expanding OO-empire. Shoot the evil OO-Spaceships, but do not hit your friendly C spaceships ...