

# C

SWE110

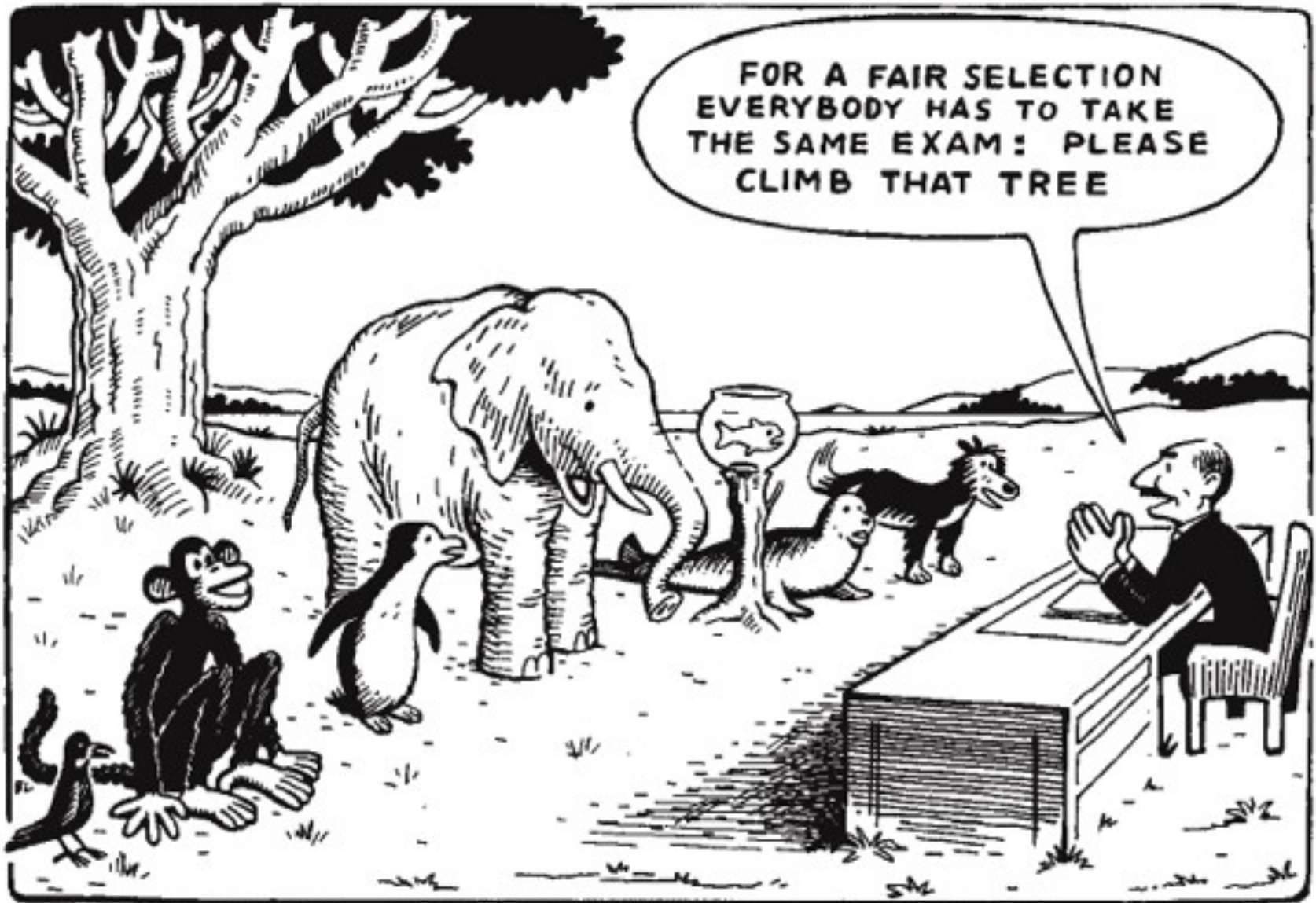
Lesson 1

Prof. Zachi Baharav

[zbaharav@cogswell.edu](mailto:zbaharav@cogswell.edu)

# Lesson 1

- In this lesson:
  - Syllabus + overview
  - Resources to help you
  - Programs:
    - “Hello world”: First program!
    - IDE, compile, link, exe.
  - Homework 0
  - Lab



Compassion over just, whenever just is not compassion.

# KISS

- Keep It Simple !!
  - Self documenting
  - Clear thinking
  - Clean code
    - Don't over use the language
  - Comment as appropriate

# Two things to note

- We'll have challenge problems every so often in the homework (hopefully in each one).
  - Everyone needs to do these.
  - They are more 'logic' exercises.
- We'll do some bigger projects together
  - Web Crawler
  - Dungeon game
  - Image manipulation



Easy to read.  
Easy to maintain.  
Modular, device independent.

Not as efficient.

C++, Java:    Objet oriented, abstracted, not machine specific.

Is integer represented as 8bit? 16bit? 64 bit?

C: High level language.

```
int    MaxLength = 5;
```

```
if (Length > MaxLength)  
    printf(" Error. Too long.");
```

Hard to read, Hard to write.

Specific!

Buggy

Efficient!

Assembler code:            load   Reg0  
                              add    Reg1  
                              store  bp01

Hardware specific machine code:   1011011  
  0110011  
  0100101

C++, Java: Object oriented, abstracted, not machine specific.

Is integer represented as 8bit? 16bit? 64 bit?

Always needs to go down to the machine code!

PreProcessor

Compile → .Obj

Link → .Exe

C: High level language.

```
int MaxLength = 5;
```

```
if (Length > MaxLength)
    printf(" Error. Too long.");
```

Assembler code:

```
load Reg0
add Reg1
store bp01
```

Hardware specific machine code:

```
1011011
0110011
0100101
```

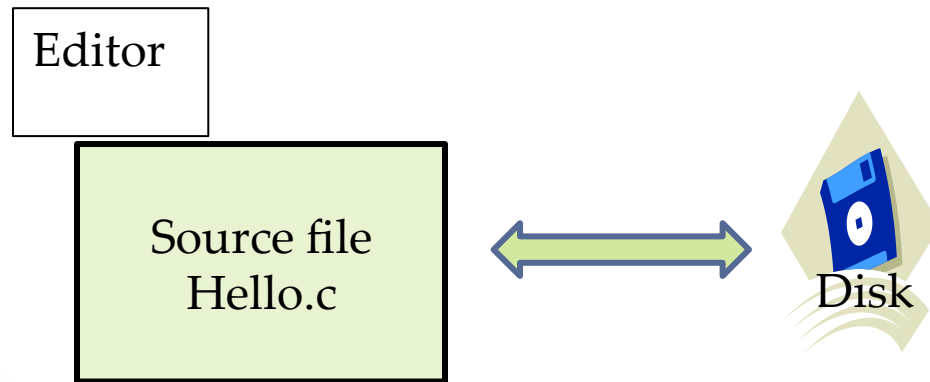
# (Dietel) 1.14 Program Development

- In general, there are 4 main steps in program development:
  - THINK !
  - Writing the program
  - Syntax : Compiling and Linking
  - Executing the program and testing the logic
- Software tools can be used with all parts of this process:
  - Microsoft Visual Studio, Eclipse, vi, emacs, ...
  - IDE – Integrated Development Environment



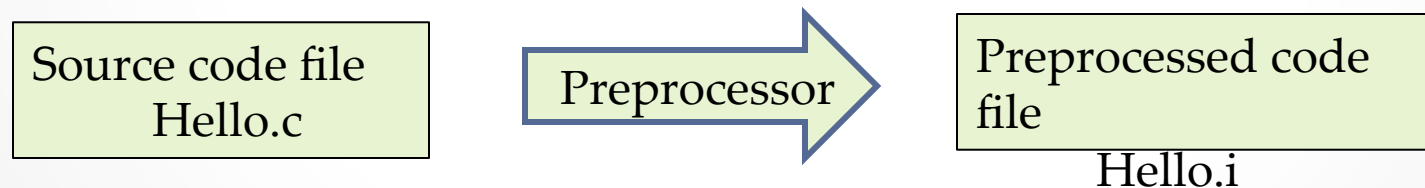
# 1.14 Program Development cont.

- Phase 1: Writing a program
  - Using a C editor, Microsoft Visual Studio, to write code in a file
  - Save the file with the .c extension



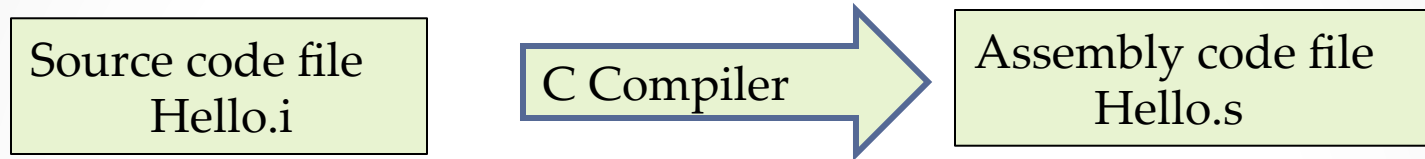
# 1.14 Program Development cont.

- Phase 2: Compile the program – includes the following four passes:
  - First pass: The `preprocessor` scan through the source code:
    - Comments are replaced with single spaces
    - Text substitute for macros
    - Include header files
    - Conditional compilations

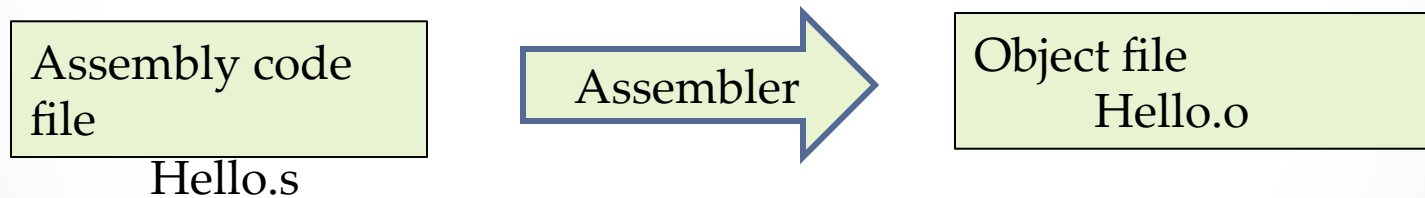


# 1.14 Program Development cont.

- Second pass: The `Compiler` turns the preprocessed file into an assembly code file



- Third pass: The `Assembler` turns the assembly source code into an assembly listing with offsets to external references called the object file.



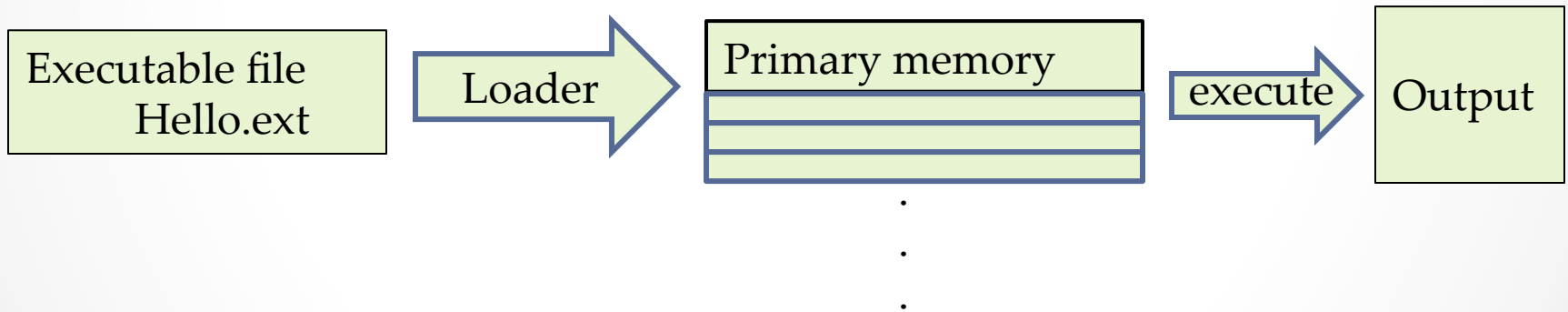
- 
- Fourth pass: The `Linker` links the object file to the external functions and variables to produce the executable file



# 1.14 Program Development cont.

- Phase 3: Execute the program

- The `loader` loads the executable file, run-time objects, modules, components from shared library into primary memory
- The computer executes the instruction sets to produce the output



# 1.14 Program Development cont.

- Most programs do not work on the first try
- Each of the phases can produce errors
- Errors detected during compiling are called **compile errors**
- Errors produced during run-time are called **run-time errors**
- Compile errors might be easier to fix as they involve **syntax errors**
- Runtime errors can be tough to find and could take lots of time to fix as they are most likely **logical errors**

# Microsoft Dreamspark

- Take the advantage of being a student to obtain a free Visual Studio
- Visit the Dreamspark website:  
<https://www.dreamspark.com/student/>
- Create an account and verify you are a student

# Teacher will do in class

- Follow along “Hello World.\n”
- Compile → .obj file
- Link (build) → .exe file
- Running from the command prompt
- Adding getchar();
- Now start on assignment 1 !
- Second lesson: Program Average.c (see assignment 1).

Time to program!!

...



# “Hello World”

- The “main()” function

- Function header
- Function body

```
#include <stdio.h>

main()
{
    printf("Hello World!\n") ;
}
```

- Preprocessor directives

- #include <stdio.h>

- printf()

The compiler is your friend!!

END