



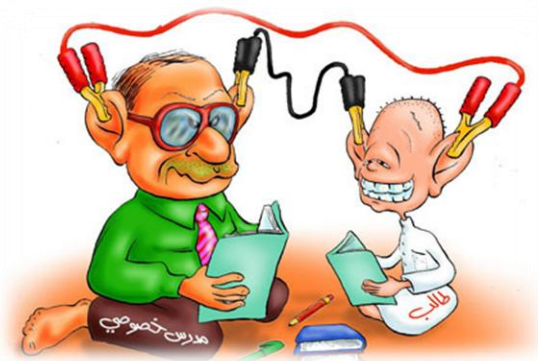
Introduction to Programming

Lecture 1: Introduction to Programming

Mahmoud El-Gayyar

elgayyar@ci.suez.edu.eg





Be a Falcon





RESPECT

*To get it,
you must give it.*



Outline

- *Introduction to the course*
 - ◆ General information
 - ◆ Syllabus
 - ◆ Course organization
 - ◆ General rules
- *Introduction to Programming*
 - ◆ Programming Skills
 - ◆ Programming Model
 - ◆ Elements of a real Programming Languages
 - ◆ History of C
 - ◆ Compiler Terminology

Outline

- *Introduction to the course*
 - ◆ General information
 - ◆ Syllabus
 - ◆ Course organization
 - ◆ General rules
- *Introduction to Programming*
 - ◆ Programming Skills
 - ◆ Programming Model
 - ◆ Elements of a real Programming Languages
 - ◆ History of C
 - ◆ Compiler Terminology

General Information

- *Lecture*

- ◆ Lecturer: Mahmoud El-Gayyar
- ◆ Contact: elgayyar@ci.suez.edu.eg
- ◆ Office hours: Monday , 11:00 am to 12:00 pm (email)

- *Lab*

- ◆ Assistant: Amira, Fatma
- ◆ Contact: ?

- *Textbooks*

- ◆ CS Department
- ◆ Many in the Library !!!


- *Course web page:*

- ◆ The notes and the assignments will be available online.
- ◆ <http://elgayyar.weebly.com/>

Syllabus of the Lecture

- *Introduction to Programming*
- *Introduction to C Language*
- *Basic Data Types*
- *Statements and Control Flow*
- *Arrays*
- *Functions and Program Structure*
- *User-defined Data Types: Structures*
- *Pointers and Memory Management*

Course Organization

- *2-3 lecture quizzes*
- *Midterm exam*
- *Assignments:*
 - ◆ Submitted in Groups: 4-6 students
 - ◆ One group will be selected to represent its solution
 - ◆ Sometimes it helps to do it twice if you feels it is still difficult
 - ◆ Deadline is a real deadline ?!! Half of the final mark 
- *Bonus marks*
 - ◆ Optional work, communication, clever or creative solutions
- *I monitor area of difficulty and adjust the material: lectures, assignments, etc.*

Additional Hints

- *Don't wait until the last minute each week*
- *If you get stuck – move around – review material, search Google, then come back*
 - ◆ *When you look back, you will see that this was all *really* easy*
- *When you feel stuck – communicate– ask a friend or come back to your assistant or to me.*

Rules !!



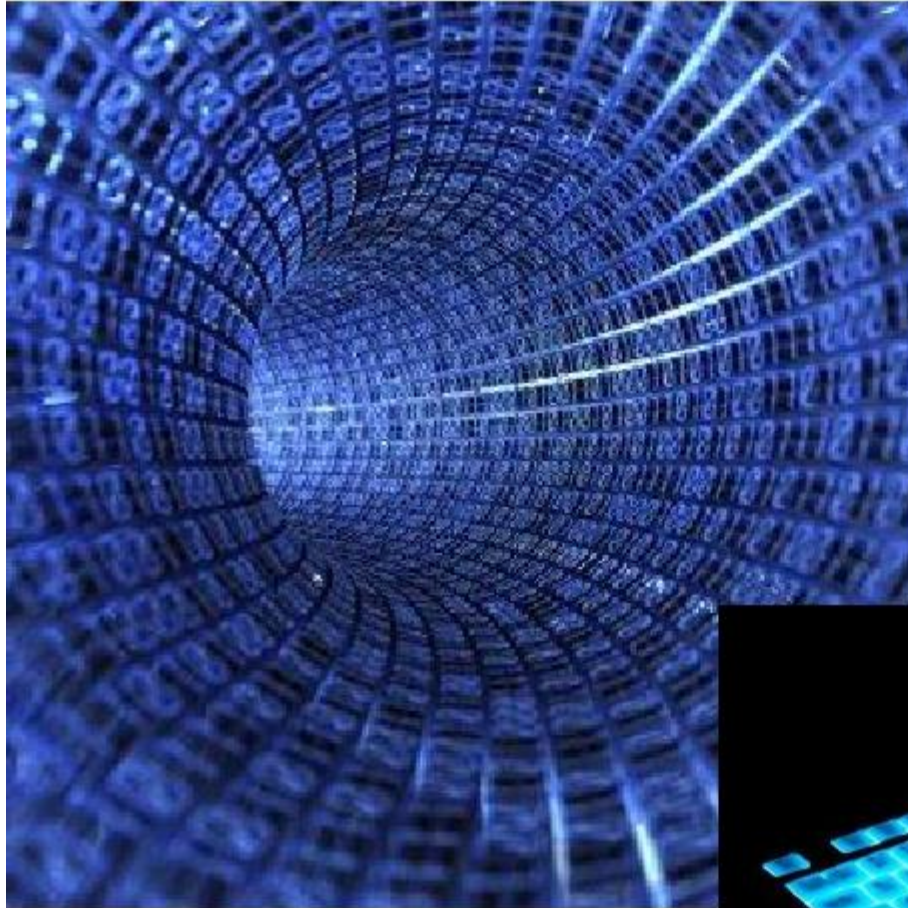
~~INTERRUPTION~~



Outline

- *Introduction to the course*
 - ◆ General information
 - ◆ Syllabus
 - ◆ Course organization
 - ◆ General rules
- ***Introduction to Programming***
 - ◆ Programming Skills
 - ◆ Programming Model
 - ◆ Elements of a real Programming Languages
 - ◆ History of C
 - ◆ Compiler Terminology

What is Programming?



Programming Skills

1. Attention to
Details



3. Good Memory



2. Stupidity

4. Abstraction



The Programming Model

- *A Computer Program:*

Dynamic



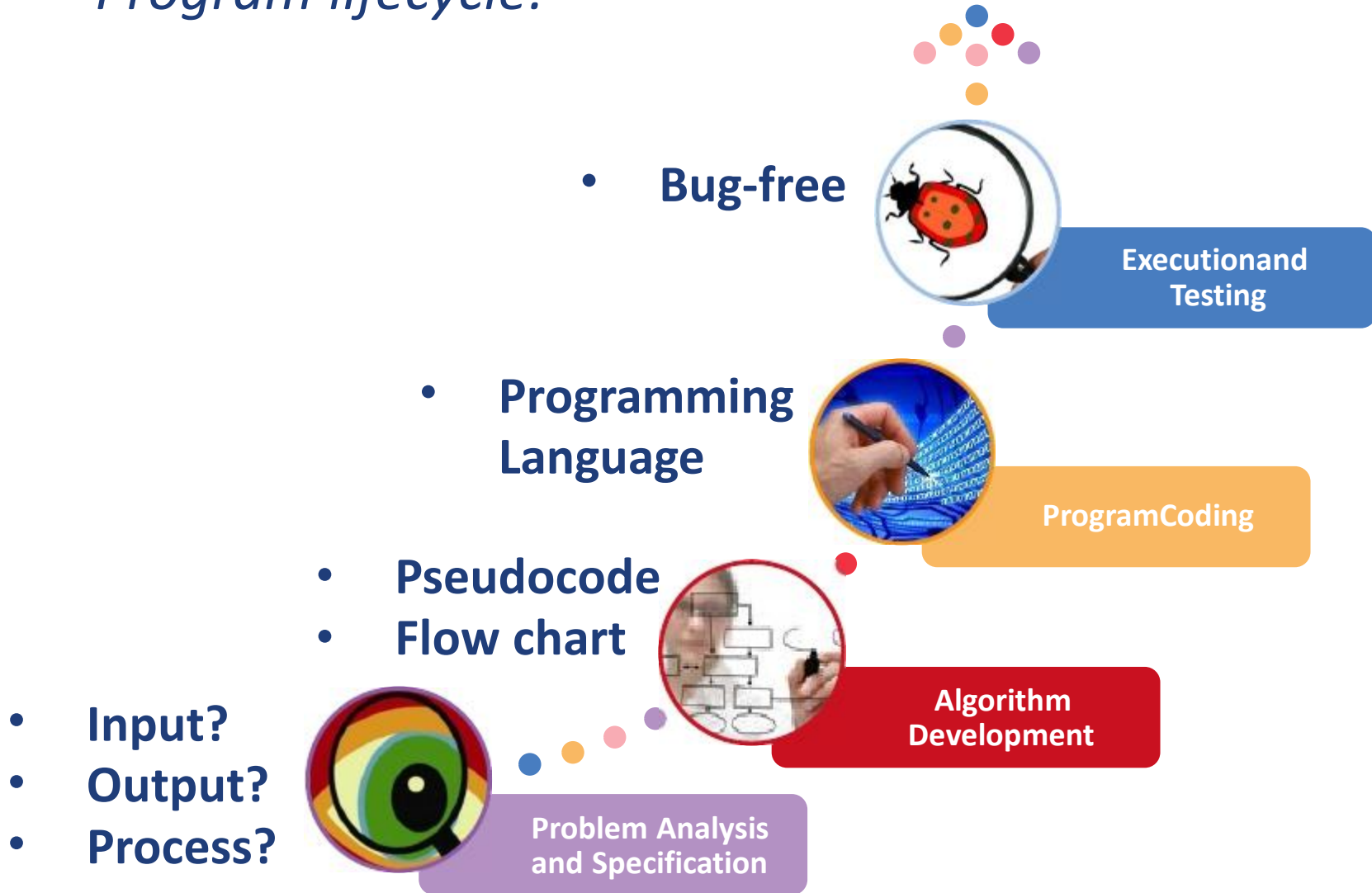
*Valuable
Information*



Static Processing

The Programming Model

- *Program lifecycle:*



Elements of a Real Programming Language

- *Where to store your data?* → *Variables (x, y, z)*
 - ◆ Memory Locations
 - ◆ Need to have certain *types*
 - *Compute new values?* → *Expressions (10+6)*
 - *Store Values?* → *Assignments(x= 10+6)*
 - *How to branch?* → *Conditionals*
 - *How to repeat?* → *Loops*
 - *How to simplify?* → *Functions*
- } Control Flow

Numbers, Characters and Strings

- *Numbers:*

- ◆ Integers (..., -2, -1, 0, 1, 2, ...)
- ◆ Float, Double (3.5, -4.002)

- *Characters: character set*

- ◆ 'A' → 65
- ◆ 'a' → 97
- ◆ '\$' → 36

- *Strings: List of characters*

- ◆ "Hello" → 'H', 'e', 'l', 'l', 'o', '\0'



Programming Languages

The Evolution of Computer Programming Languages

```
4b 54 68 64 00 00 00 04 00
72 68 00 00 00 61 00 f0 0a
00 41 f7 00 00 00 00 00 c0
5a 32 01 00 00 00 00 00 00
00 00 00 ff 51 03 04 0a 18
5a 6f 06 00 00 43 40 2c 00
00 42 00 04 00 41 5e 2c 00
00 40 00 00 ff 2f 00 45 54
88 00 00 00 c5 04 00 ff 79
00 00 00 00 00 10 00 00 00
03 07 c2 22 00 69 62 23 0f
74 00 01 00 00 00 00 00 00
00 00 00
```



Hex

```
0000          0000 0000
Mvz2         00  1
SIx2B        001  0x1,0
COvzT        001  0x1,0

0000          000  00
ALvz          000  0x00
0000          000  0x00
0000          000  0x00
0000          000  0x00
0000          000  0x00
0000          000  0x00
0000          000  0x00
0000          000  0x00
0000          000  0x00
0000          000  0x00
```



Assembler

```
#include <stdio.h>
#include <io.h>
#include <dos.h>
#include <string.h>
#include <time.h>

main()
{
char ch,*st;
int hit;
int done;
FILE *f;
system("cls");
printf("Enter a string:");
scanf("%s",st);
printf("The string is: %s\n",st);
return 0;
}
```



C

```
int main()
{
char str[100];
int i;
for(i=0; i<100; i++)
{
str[i] = 'A';
}
return 0;
}
```



C++

```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<String> list = new ArrayList<>();
        while (scanner.hasNext()) {
            list.add(scanner.nextLine());
        }
        Collections.sort(list);
        for (String s : list) {
            System.out.println(s);
        }
    }
}
```



Java

Low

Middle

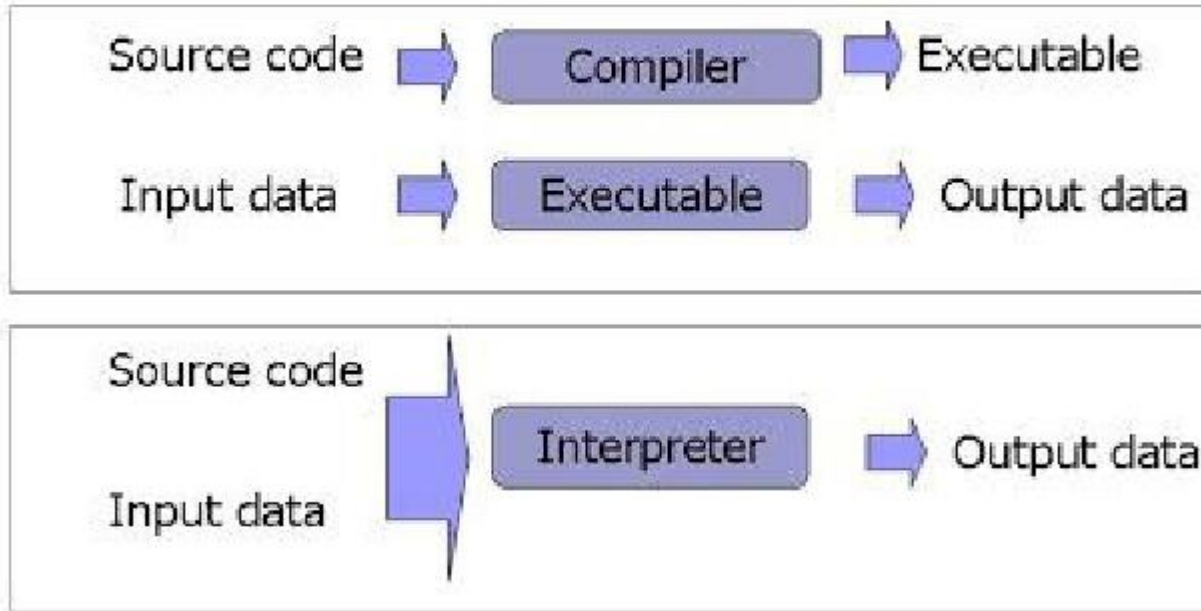
High

History of C

- *Dennis Ritchie (American) created C in 1972*
 - ◆ He also one of the Unix operatingsystem designers
- *A tool for programmers that needed:*
 - ◆ More readable language than Assembly
 - ◆ Still have low level access capabilities
- *C is best for:*
 - ◆ The programs that must run quickly
 - ◆ Need low level access to computer hardware (e.g. need direct control of I/O or memory)

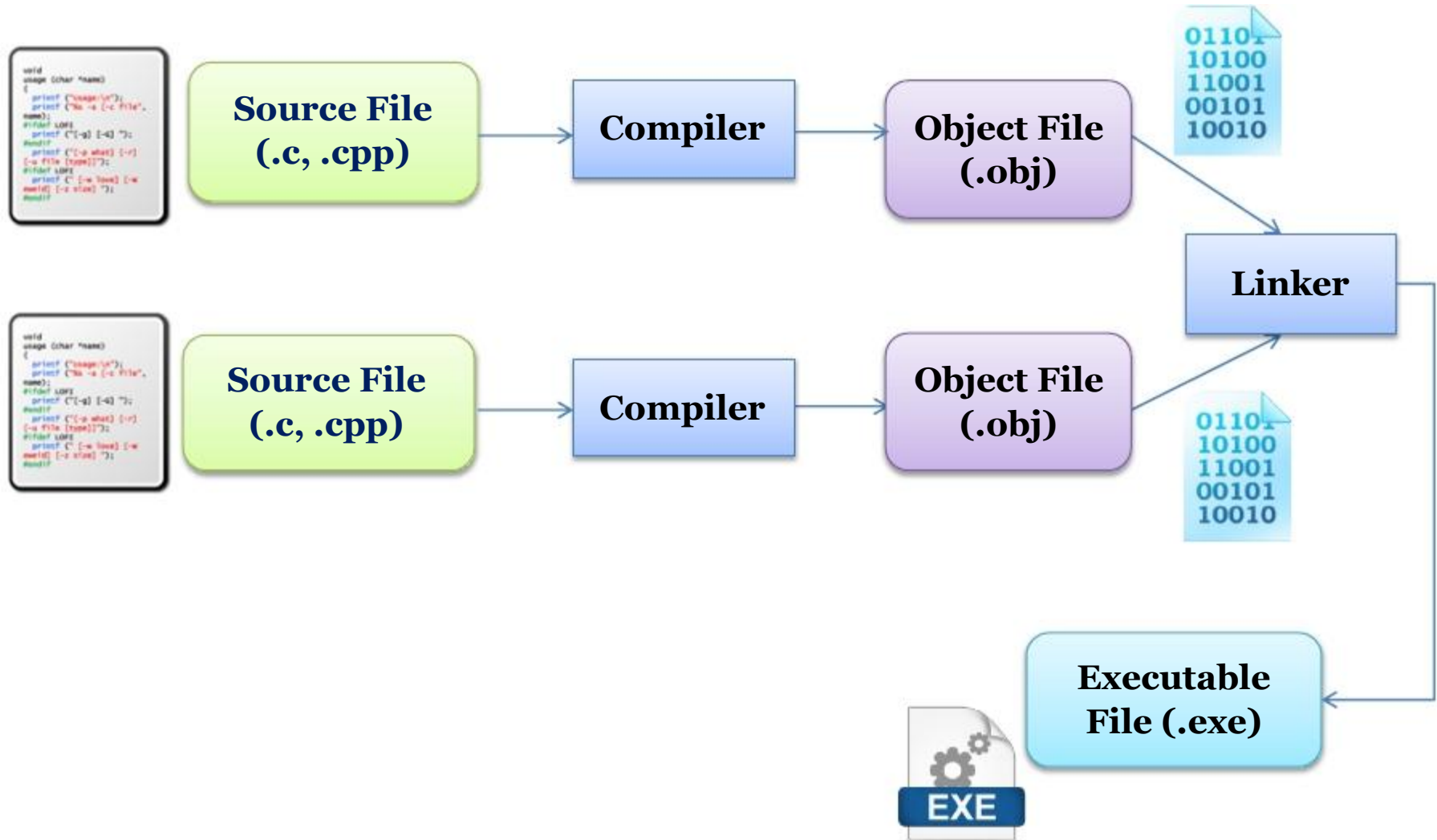
Compiler Terminology

Ideal concept:



- *Convert your source code into machine-language executable programs.*
 - ◆ Compiler (e.g. C) or Interpreter (Basic)
- *Interpreter*
 - ◆ Line by line and result available immediately → No executable
 - ◆ To run, need the interpreter (in general slower than the compiler)

Compiler Terminology



Summary

- *Introduction to the course*
 - ◆ Always check the course webpage
 - ◆ Again, don't forget the deadline rule
- *Introduction to programming*
- *What is C Language?*
- *Compiler vs. Interpreter*