PROGRAMMING FOR COMPUTATIONAL CHEMISTRY

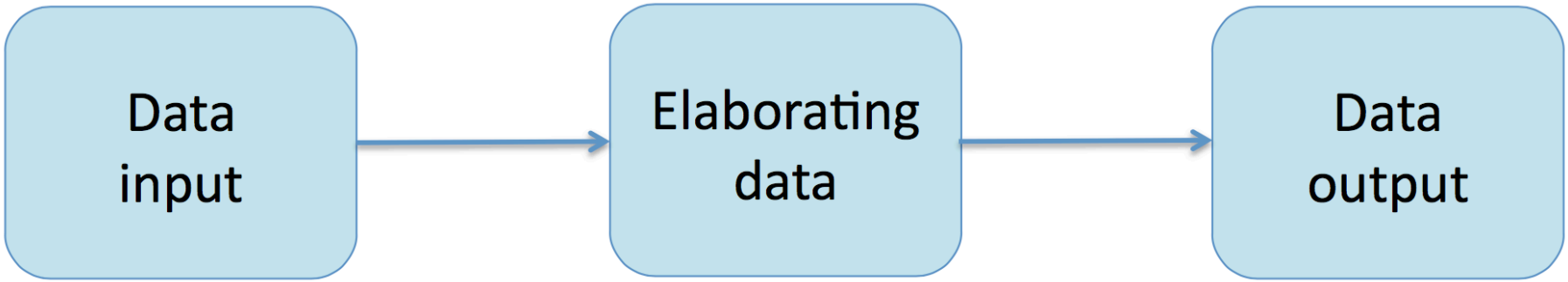
Introduction to Fortran

Emanuele Coccia

Dipartimento di Scienze Chimiche e Farmaceutiche

General flowchart

Goal for a computer: executing a set of instructions to solve a given numerical problem



Programming language

- Reproducible syntax to write instructions for a computer

Programming language

- Reproducible syntax to write instructions for a computer
- Separation between the physical support and instructions

Programming language

- Reproducible syntax to write instructions for a computer
- Separation between the physical support and instructions
- Semantics: assign a meaning to the syntactic forms of the language

Top-down design

- Computer Science: best method to translate a scientific model into a code

Top-down design

- Computer Science: best method to translate a scientific model into a code
- Start from the general instructions: input, do operations, output

Top-down design

- Computer Science: best method to translate a scientific model into a code
- Start from the general instructions: input, do operations, output
- From the general problem to the identification of the main sections

Top-down design

- Computer Science: best method to translate a scientific model into a code
- Start from the general instructions: input, do operations, output
- From the general problem to the identification of the main sections
- Simplify towards smallest pieces of code (stepwise refinement)

Define an algorithm

- To solve a problem, the proposed approach has to be:

Define an algorithm

- To solve a problem, the proposed approach has to be:
  1. clear and univocally defined

Define an algorithm

- To solve a problem, the proposed approach has to be:
  1. clear and univocally defined
  2. effective

Define an algorithm

- To solve a problem, the proposed approach has to be:
  1. clear and univocally defined
  2. effective
  3. finite

Define an algorithm

- To solve a problem, the proposed approach has to be:
  1. clear and univocally defined
  2. effective
  3. finite
- Two steps:

Define an algorithm

- To solve a problem, the proposed approach has to be:
  1. clear and univocally defined
  2. effective
  3. finite
- Two steps:
  1. Develop or choose an algorithm

Define an algorithm

- To solve a problem, the proposed approach has to be:
  1. clear and univocally defined
  2. effective
  3. finite
- Two steps:
  1. Develop or choose an algorithm
  2. Coding the algorithm

Compiling a code

- From high- to low-level (machine) language: the role of the compiler

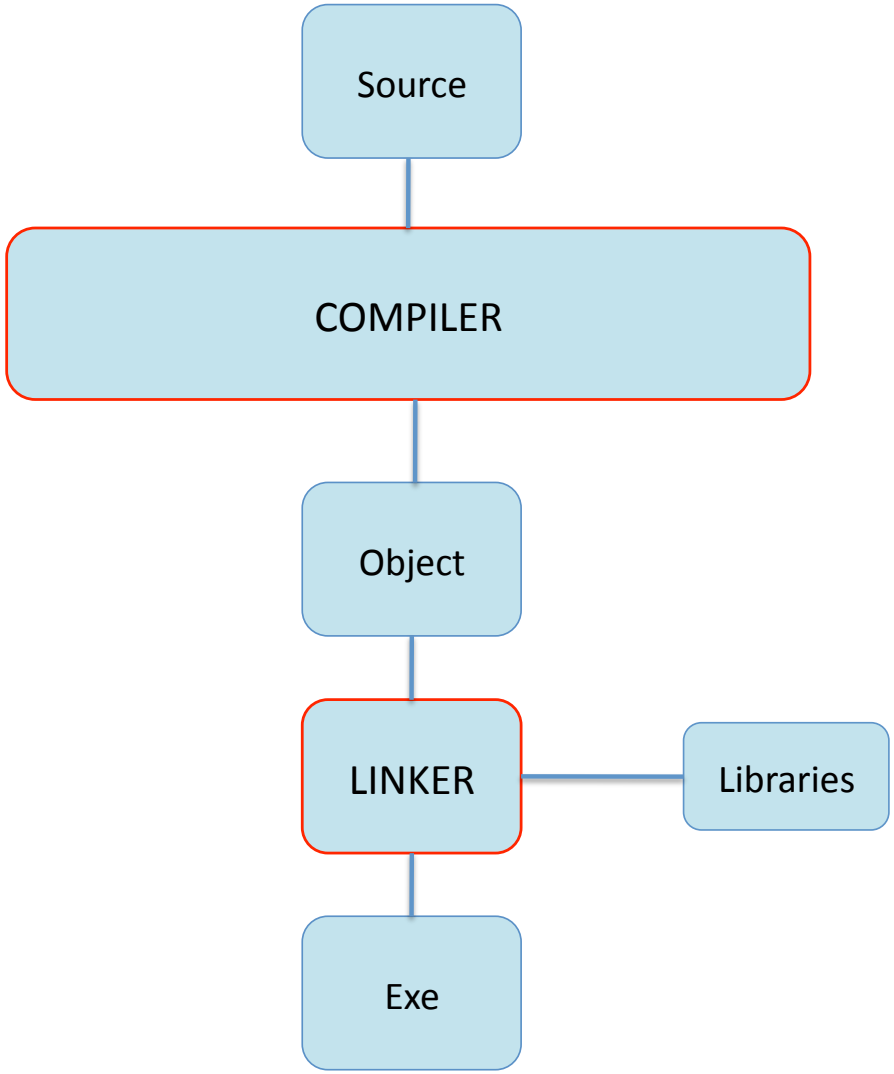
Compiling a code

- From high- to low-level (machine) language: the role of the compiler
- From a source code (e.g., name.f90) to an executable

Compiling a code

- From high- to low-level (machine) language: the role of the compiler
- From a source code (e.g., name.f90) to an executable
- Two main steps:
  1. Translating the source code into an object file, containing meta-instructions
  2. Converting the object (and possibly other ones in libraries) into an executable

Compiling a code



Errors in programming

- Syntax errors: recognised by the compiler and (usually) easy to fix

Errors in programming

- Syntax errors: recognised by the compiler and (usually) easy to fix
- Semantic errors: use debugging options to find and fix them, structured programming to avoid them

Fortran

- First high-level language

Fortran

- First high-level language
- Largely used for scientific and technical purposes

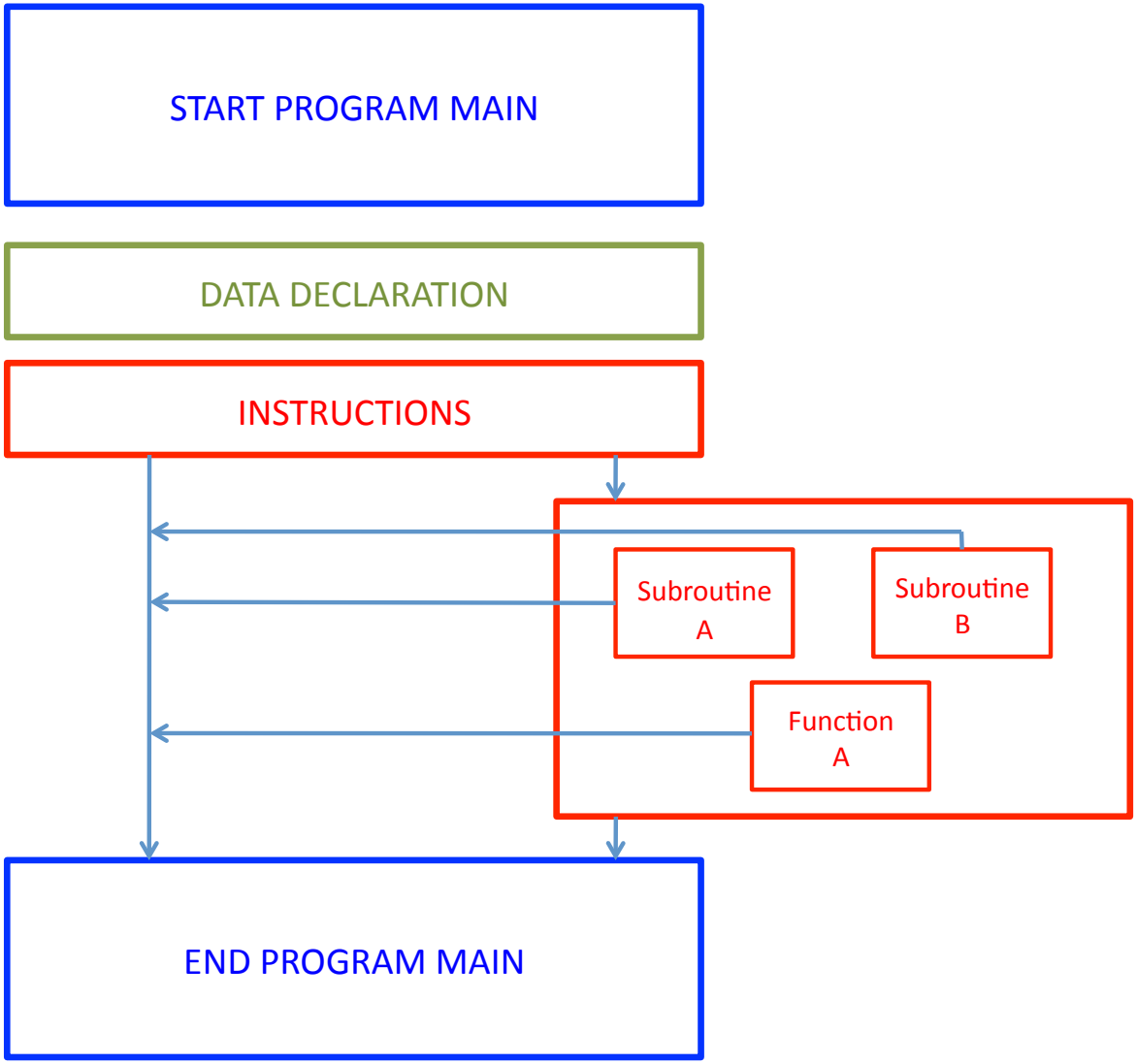
Fortran

- First high-level language
- Largely used for scientific and technical purposes
- Born in the mid-50s: FORmula TRANslator

Fortran

- First high-level language
- Largely used for scientific and technical purposes
- Born in the mid-50s: FORmula TRANslator
- Standard releases:
  - FORTRAN4
  - FORTRAN66
  - FORTRAN77
  - Fortran90
  - Fortran95
  - Fortran03
  - Fortran08
  - ...

Generic (Fortran) code



Our first code

- Open a terminal

Our first code

- Open a terminal
- Type vi hello_world.f90

Our first code

- Open a terminal
- Type vi hello_world.f90
- Let us do it together (hello_world.f90)

Compiling and running

- ifort -c hello_world.f90 $\rightarrow$ produces hello_world.o

Compiling and running

- ifort -c hello_world.f90 $\rightarrow$ produces hello_world.o
- ifort -o hello_world.x hello_world.f90 $\rightarrow$ produces the executable hello_world.x

Compiling and running

- ifort -c hello_world.f90 $\rightarrow$ produces hello_world.o
- ifort -o hello_world.x hello_world.f90 $\rightarrow$ produces the executable hello_world.x
- (Some) compiler options:

Compiling and running

- ifort -c hello_world.f90 $\rightarrow$ produces hello_world.o
- ifort -o hello_world.x hello_world.f90 $\rightarrow$ produces the executable hello_world.x
- (Some) compiler options:
  1. -o object (executable)

Compiling and running

- ifort -c hello_world.f90 $\rightarrow$ produces hello_world.o
- ifort -o hello_world.x hello_world.f90 $\rightarrow$ produces the executable hello_world.x
- (Some) compiler options:
  1. -o object (executable)
  2. -c compile only

Compiling and running

- ifort -c hello_world.f90 $\rightarrow$ produces hello_world.o
- ifort -o hello_world.x hello_world.f90 $\rightarrow$ produces the executable hello_world.x
- (Some) compiler options:
  1. -o object (executable)
  2. -c compile only
  3. -pg profiling

Compiling and running

- ifort -c hello_world.f90 $\rightarrow$ produces hello_world.o
- ifort -o hello_world.x hello_world.f90 $\rightarrow$ produces the executable hello_world.x
- (Some) compiler options:
  1. -o object (executable)
  2. -c compile only
  3. -pg profiling
  4. -On optimisation level

Compiling and running

- ifort -c hello_world.f90 $\rightarrow$ produces hello_world.o
- ifort -o hello_world.x hello_world.f90 $\rightarrow$ produces the executable hello_world.x
- (Some) compiler options:
  1. -o object (executable)
  2. -c compile only
  3. -pg profiling
  4. -On optimisation level
  5. ...

Source and binaries

- Two types of files: binary or text
  - Binaries can be programs or data files written by programs or the operating system
  - Text files are those that can be read by us human beings
- Two categories of files: programs or data
- Either a file contains a set of instructions to be executed by the CPU (a program) or it contains information (data)