

Computing for Musicology

(Course code: F104N5)

0. Introduction to the course

J.N. Oliveira

Dept. Informática,
Universidade do Minho
Braga, Portugal

March 2009 (last update: September 2025)

Licenciatura em Música
Universidade do Minho
Braga

Informatics...

Informatics, computers... What is this for?

- A **computer** is a machine that does very **laborious** things for us...
- ... **very quickly** ...
- ... without getting **annoyed** ...
- ... and without making **mistakes**!



Informatics...

Informatics, computers... What is this for?

- A **computer** is a machine that does very **laborious** things for us...
- ... **very quickly** ...
- ... without getting **annoyed** ...
- ... and without making **mistakes**!



Computing — hardware, software, ...

Computing:

- **Hardware** — the physical machine itself
- **Software** — the tools, programs, applications which run on top of the hardware

Questions:

- What is **software**?
- Where do **programs** come from?
- Programming: **invention**? **construction**?

Computing — hardware, software, ...

Computing:

- **Hardware** — the physical machine itself
- **Software** — the tools, programs, applications which run on top of the hardware

Questions:

- What is **software**?
- Where do **programs** come from?
- Programming: **invention**? **construction**?

Software: where do programs come from

There are a number of misconceptions concerning computer programming, in particular:

- Programming is (very) **difficult**
- Only bright people can **program** a computer
- Programming is sheer “**art**” — you have to be one of the elected few who understand it...

Teaching programming

Nonsense! It turns out that

- Computer **programs** emerge from ordinary mathematics
- Teaching **computer programming** could start around K12, if not earlier, as an activity close to mathematics training



And it is analogous to **composing music**.



Teaching programming

Nonsense! It turns out that

- Computer **programs** emerge from ordinary mathematics
- Teaching **computer programming** could start around K12, if not earlier, as an activity close to mathematics training

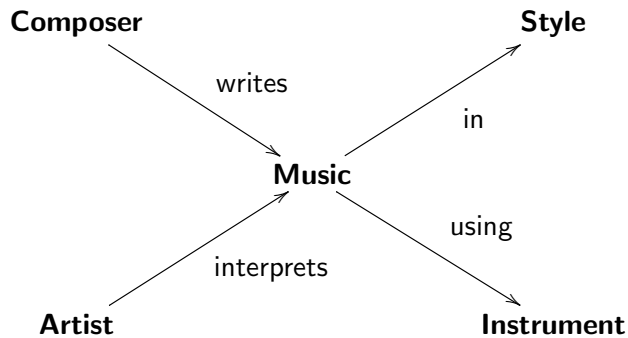


And it is analogous to **composing music**.



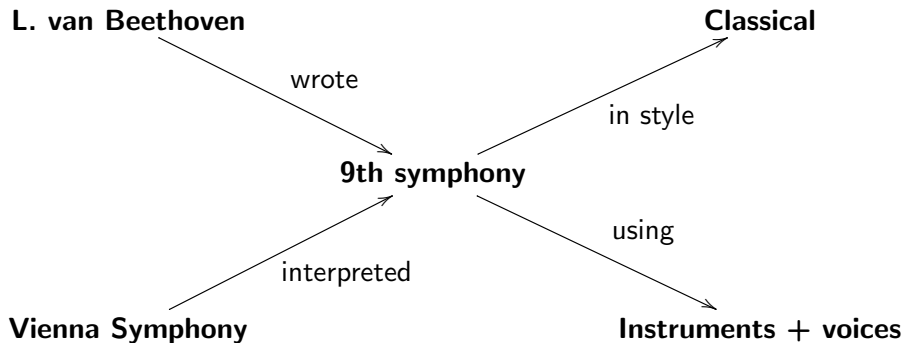
What does it mean to program?

Analogy:



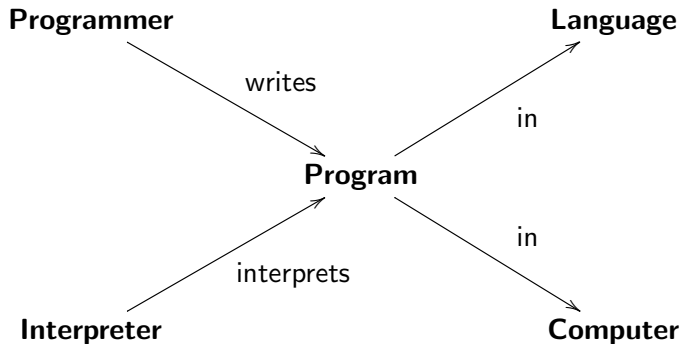
Computer programming

Example:



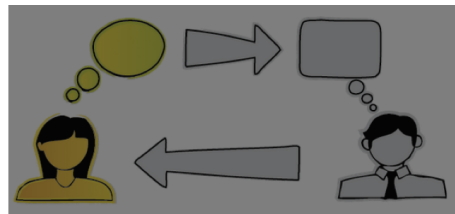
Computer programming

Programming:



Languages?

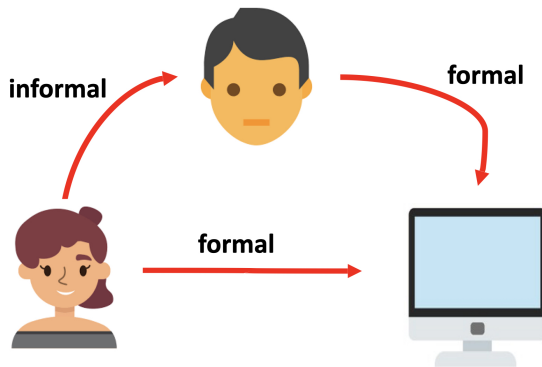
Humans talk to each other using natural **languages**.



Talking to a **machine** is not very different — we also need "languages".

Kinds of Language

But such languages
need to be
understandable by
machines.



Kinds of Language

We need languages

- to describe **objects**
- to give the machines **instructions** for them to perform **actions** which we regard as useful.

Thus the classification:

- Domain specific languages (**DSLs**) — which describe **objects**, eg. **music**, **text**, **videos**, **web sites** and so on
- **Programming languages** — which instruct machines how to replace humans and perform **actions**.

DSLs

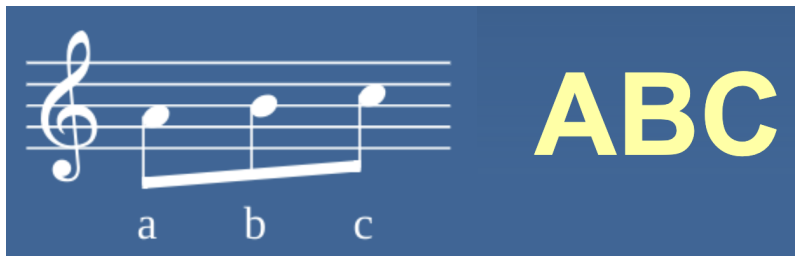
We shall get in touch with the following DSLs/systems:

- **ABC** — for describing music
- **MarkDown** — for describing websites
- **LaTeX** — for describing text
- **OpenShot** — for describing videos

Everything will be web-based: no need for installing anything.

ABC: a DSL for denoting music

Concerning **music**, we will resort to one called



Programming

Concerning **programming**, we will resort to one called

Haskell



available from the **Jupyter** server at DIUM.

Programming

Let us try it!



Numbers

We could see that it all behaves like an ordinary **calculator** as far as **numbers** are concerned.

Is this all Haskell and Jupyter have to offer?

No...

