





Subjects	Goals
Try and Catch	<ul style="list-style-type: none"><li>• Use Try and Catch to control the flow of errors in the program</li></ul>
Exercises/Review	<ul style="list-style-type: none"><li>• complete exercises given in class</li></ul>

# Understanding fundamental concepts

1. A **language** is a means (and a tool) for expressing and recording thoughts.
2. A **natural language** is a language people use to communicate with each other in everyday life. English, Russian, German, Swahili, and Hindi are examples of natural languages.
3. A **programming language** is a language developed by humans and used to communicate with computers. A programming language has a set of means to instruct a computer what to do and how.
4. A **high-level programming language** is a programming language which operates on a high level of abstraction thereby allowing the developer to ignore the physical details of the computer's hardware, for example the CPU type, memory size and organization, etc. Python, JavaScript, and C/C++ are all examples of high-level programming languages.
5. A **machine language** is a language placed at the lowest level of computer programming. It's a sequence of bits (binary digits usually recognized as **0** and **1**) which directly forces the CPU to execute the desired elementary operations.
6. An **instruction list** (abbreviated to **IL**) is a list of all elementary (atomic) operations which can be executed by a certain CPU. For example, **x86** (used in personal computers) and **arm** (used in mobile devices) processors have different and incompatible instruction lists).
7. A **source code** is a text encoded in any of the programming languages (regardless of the language's level). Usually, the source code is put inside a text file which resides inside the developer's computer filesystem, while the file name's extension reveals the programming language used to write the code (for example, files with names which ends with `.py` contain Python source code, while the `.cpp` extension marks files which hold C++ (usually pronounced as *see-plus-plus*) source code.

8. Any language (no matter if it's natural or artificial) is constituted by:
  - an **alphabet** understood as a set of symbols used to build words of a certain language (e.g. the Latin alphabet for English, Cyrillic alphabet for Russian, Kanji for Japanese, and so on).
  - a **lexis**, also known as a **dictionary**, is a set of words the language offers its users (for example, the word "chat" is present both in English and French dictionaries, but its meaning is obviously different).
  - **syntax** is a set of rules used to determine if a certain sequence of words forms a valid sentence.
  - **semantics** is defined as a set of rules which settle whether or not a certain phrase or sentence makes sense in a given language.
9. A source code **cannot be directly executed by a computer**. To make it possible the source code has to be **translated** into a machine code accepted by a target computer and its CPU. This task can be done using two different techniques:
  - **compilation** performed by a one-time translation of the source program; an executable binary file is created in effect – the file can be run at any time without the need to have the source code; the program that performs the above translation is called a **compiler** or **translator**.
  - **interpretation** which involves a dedicated program designed to translate the source program on-the-fly each time it has to be run; the program performing this task is called an **interpreter**; this means that the interpreter is needed whenever the source code has to be executed.
10. A specific programming language is designed to be the object of either compilation or interpretation (this choice imposes certain distinctive features onto the language). For example, **Python** is an **interpreted** programming language, while **C++** is a compiled one.
1. The interpreter and its environment, created and distributed by the **Python Software Foundation** (PSF) is written mostly in the programming language. It allows Python to be easily ported and migrated to all platforms providing the ability to compile and run language programs. This is also why the PSF implementation is often referred to as **CPython**. CPython is the most widely used implementation of Python.



BIAS

