A Gentle Introduction to Programming

Erik Zeiner

Fachschaft General & Computational Linguistics
University of Tübingen

WS 2024/25
Pre-course

What is programming?

Providing instructions on how to perform a task

- This can take many forms; just look at the history of computers!
- Nowadays, we are usually talking about computer programming - writing code

Programming vs. Algorithms

Programming

A sequence of instructions used to tell the computer to perform a task.

Algorithm

A sequence of steps used to solve a specific problem.

Within the scope of programming, you will be both:

- using existing algorithms and
- designing your own

How does the computer follow your instructions?

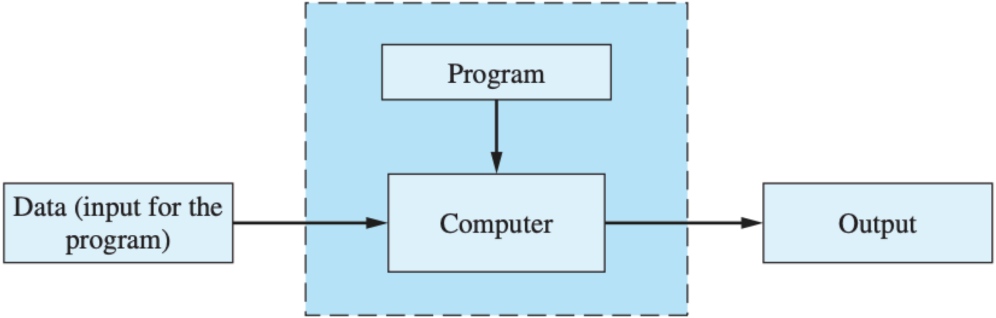
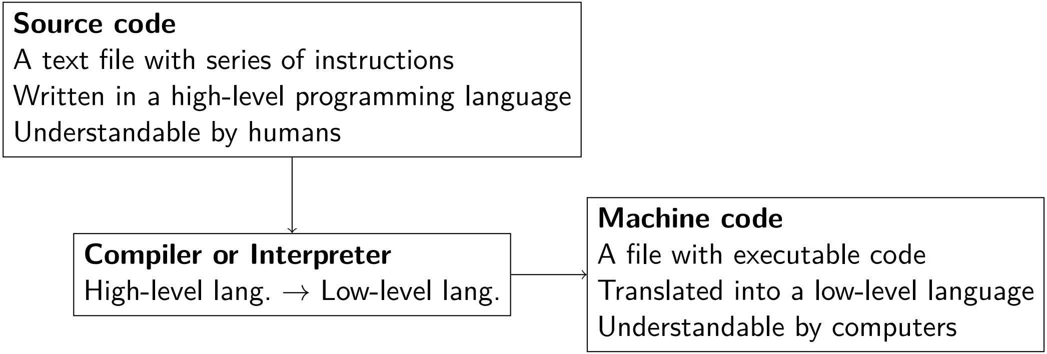


Abbildung: From 'Java: An Introduction to Problem Solving & Programming'

How does the computer understand your code?



Source code

A text file with series of instructions

Written in a high-level programming language

Understandable by humans

Compiler or Interpreter

High-level lang. → Low-level lang.

Machine code

A file with executable code

Translated into a low-level language

Understandable by computers

How do you actually go about writing a program?

Theoretical part

When faced with a real-world problem we want to solve:

- Analyse the problem - understand it as a series of discrete logical steps
- Design a way to solve it - again, thinking in steps
  This could be done on paper; it's not about coding, it's about thinking!
- Instruct the computer to follow your solution

How do you actually go about writing a program?

Practical part

1. **Write code** - *it's just plaintext files*
   text editors provide helpful tools and features for writing code
2. **Run & Test code** - text files interpreted/compiled and instructions executed
   Use the command line to tell the computer to run the source file
3. **Debug code** - find and correct errors
   manually or with a debugger

Repeat 1. - 3.

- **Tools:** countless apps, workflows, and ways of writing code - personal preferences + imposed requirements
  IDEs - write, run, and debug in one app

How does one choose a programming language?

- Personal preference/knowledge
- Imposed requirements - platform, project, group, or a company standardisation
- Advantages - use-case, speed, available libraries/packages,...

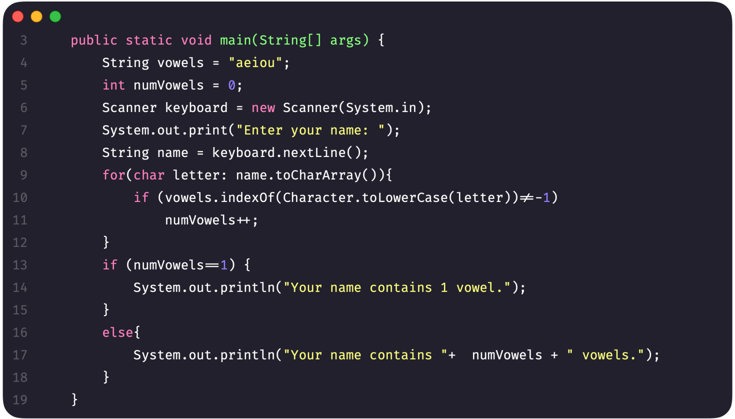
Concepts and skills are highly transferable from one language to another

knowing any programming language means others will be easier to pick up

Java

Classes you will use it in include: DSA I, DSA II

- Fast
- Compiled
- Platform independent
- Object oriented
- Quite in-demand by employers
- A tad wordy

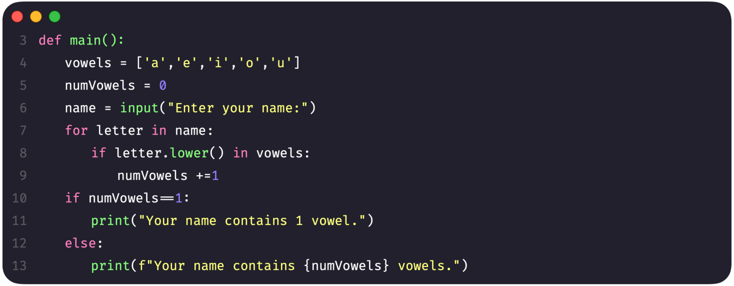


```java
public static void main(String[] args) {
    String vowels = "aeiou";
    int numVowels = 0;
    Scanner keyboard = new Scanner(System.in);
    System.out.print("Enter your name: ");
    String name = keyboard.nextLine();
    for(char letter: name.toCharArray()){
        if (vowels.indexOf(Character.toLowerCase(letter))!=-1)
            numVowels++;
    }
    if (numVowels==1) {
        System.out.println("Your name contains 1 vowel.");
    }
    else{
        System.out.println("Your name contains "+  numVowels + " vowels.");
    }
}
```

Python

Classes you will use it in include: Programming and Data Analysis, DSA III

- Not as fast but more versatile
- Interpreted
- Platform independent
- Object oriented
- Quite in-demand by employers
- Simpler Syntax



```python
def main():
    vowels = ['a','e','i','o','u']
    numVowels = 0
    name = input("Enter your name:")
    for letter in name:
        if letter.lower() in vowels:
            numVowels +=1
    if numVowels==1:
        print("Your name contains 1 vowel.")
    else:
        print(f"Your name contains {numVowels} vowels.")
```

ASK A QUESTION

OUR WEBSITE