1DV532 – Starting Out with Java

Lesson 1

Introduction to Computers, Programming and Problem Solving

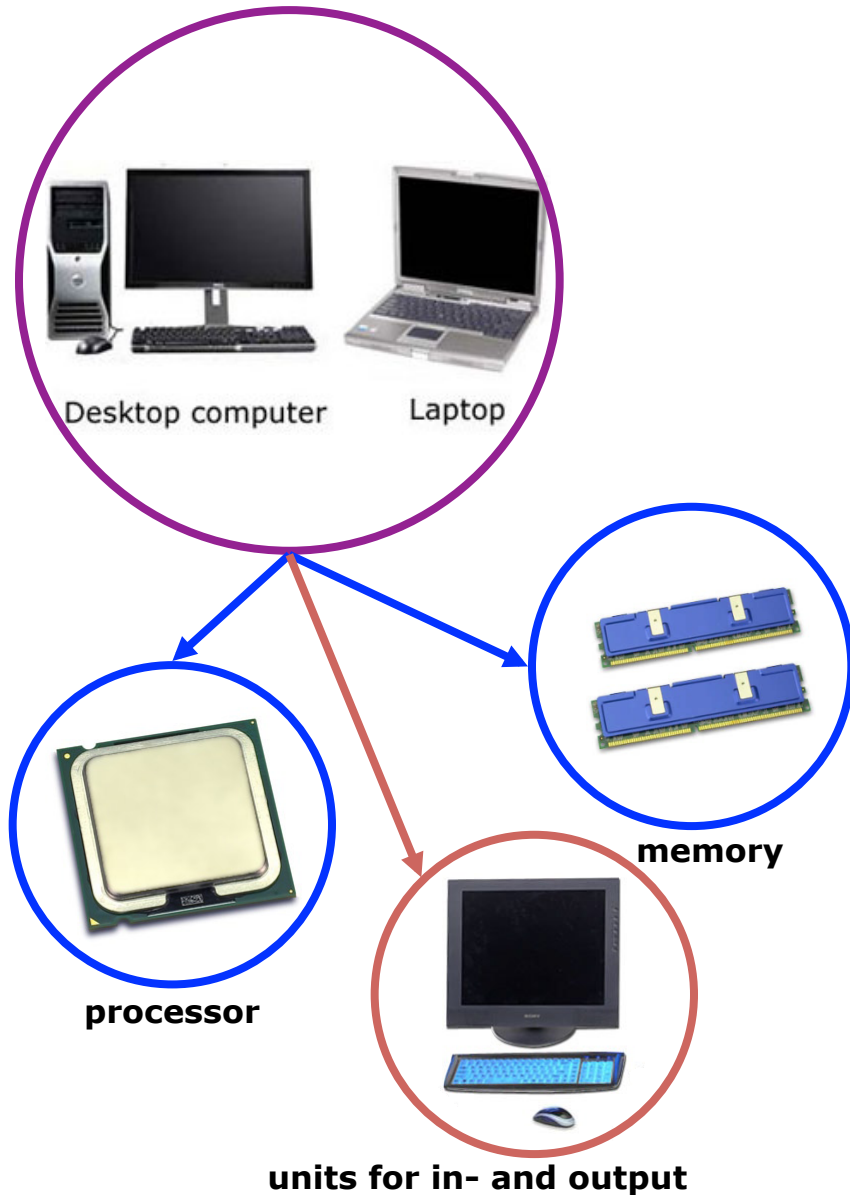Dr. Nadeem Abbas

nadeem.abbas@lnu.se

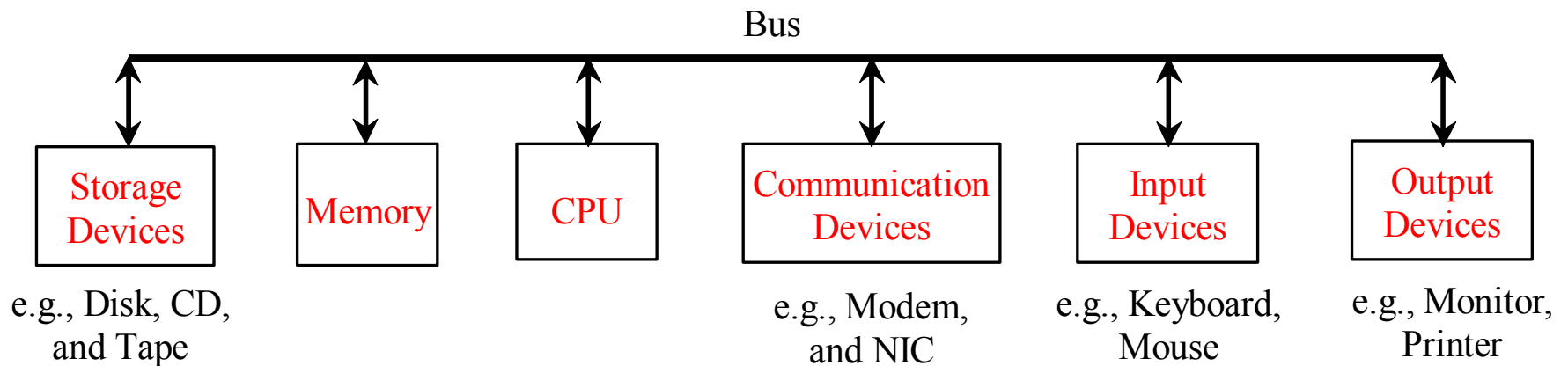# Introduction to Computers

What is a Computer?

Desktop computer   Laptop

processor

memory

units for in- and output

- **A computer** is an electronic machine that can store and process data.
- A computer mainly consists of three main components:
    1. **Processor**,
    2. **Memory**,
    3. **Input/Output (I/O) units**

- **The processor**, or the microprocessor, is also known as the *Central Processing Unit*, **CPU**.
- It is the processor that **performs the instructions stored in the memory**.

- **A set of instructions** that instructs the processor **is called computer program**.
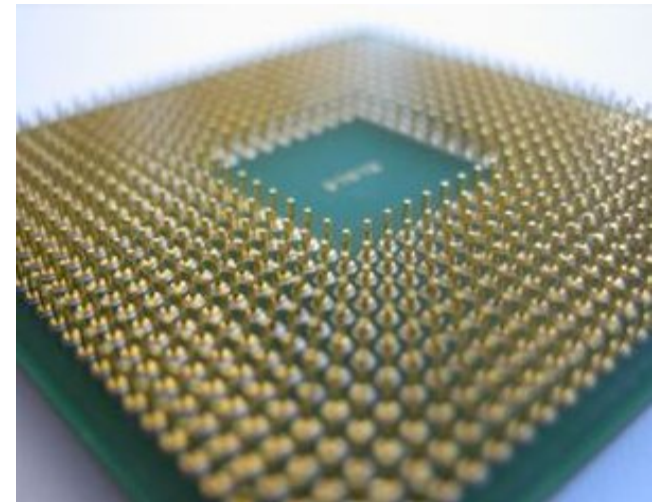
What is a Computer?

In addition to CPU, memory and I/O devices, a computer involves few other devices such as, hard disk, floppy disk, printer, and communication devices.

Bus

Storage Devices	Memory	CPU	Communication Devices	Input Devices	Output Devices

e.g., Disk, CD, and Tape

e.g., Modem, and NIC

e.g., Keyboard, Mouse

e.g., Monitor, Printer

Linnæus University

Central Processing Unit (CPU)

- The central processing unit (CPU) is the brain of a computer. It retrieves instructions from memory and executes them.

- A most common measure of CPU speed (performance) is the clock speed, which is measured in Mega or Giga Hertz.

  – 1 MegaHertz (MHz) = 1 million pulses per second.

  – 1 Giga Hertz (GHz) = 1000 MHz

- Higher the clock speed, the more operations the CPU can execute per second.

Memory (Primary Storage)

- *Memory* is to store **data** and **program instructions** for CPU to execute.

- A program and its data must be brought to memory before it can be executed.

- A memory unit is an ordered sequence of **bytes**,

  - **1 byte** = **8 bits**.

- A memory byte is never empty, but its initial content may be meaningless to your program.

- The current content of a memory byte is lost whenever new information is placed in it.
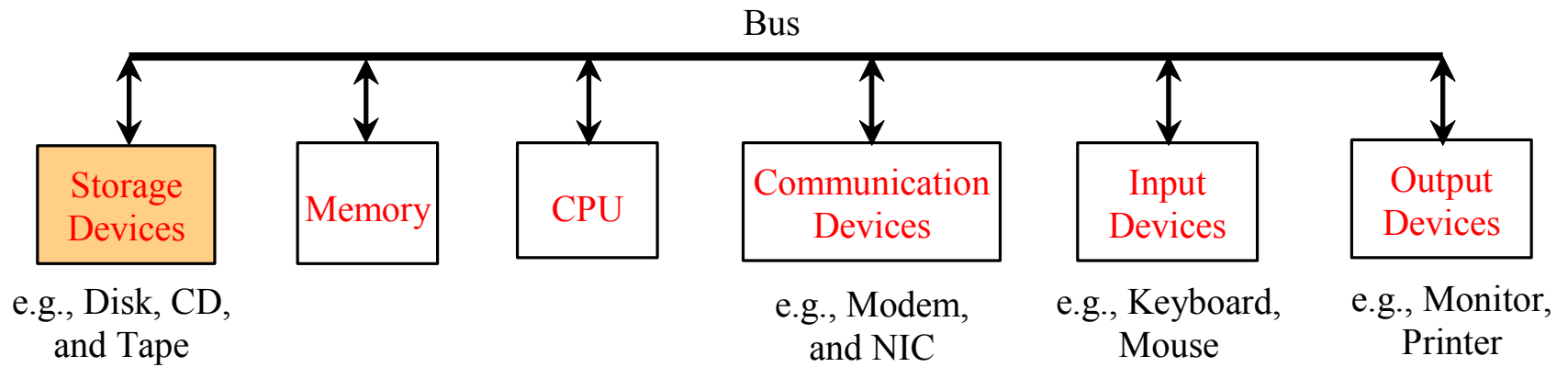
How Data is Stored?

- Data of various kinds, such as numbers, characters, and strings, are ***encoded and stored*** as a **series of bits** (zeros and ones).

- Computers use zeros and ones because digital devices have two stable states, which are referred to **as *zero* and *one*** by convention.

- The programmers need not to be concerned about the encoding and decoding of data, which is performed automatically by the system based on the encoding scheme.

- If computer needs to store a large number that cannot fit into a single byte, it uses a number of adjacent bytes.

- A byte = 8 bit is the minimum storage unit.

Memory address	Memory content	
.	.	
.	.	
.	.	
2000	01001010	Encoding for character 'J'
2001	01100001	Encoding for character 'a'
2002	01110110	Encoding for character 'v'
2003	01100001	Encoding for character 'a'
2004	00000011	Encoding for number 3

Linnæus University

Secondary Storage Devices

- Memory is volatile, because **data stored in memory gets lost when the power is off.**

- We need Secondary Storage Devices such as Hard disk, CD, DVD, Magnetic Tapes, etc, to store programs and data permanently.

- CPU, however, cannot access data directly from secondary storage devices, the program and data stored on secondary storage need to move to the memory (primary storage) for processing by CPU.

Bus

| Storage Devices | Memory | CPU | Communication Devices | Input Devices | Output Devices |

e.g., Disk, CD, and Tape

e.g., Modem, and NIC

e.g., Keyboard, Mouse

e.g., Monitor, Printer

Linnæus University

Computer Programming

What is a (Computer) Program?

- A **program** is a **set of instructions** given to a computer to solve a problem or to perform a specific task.

- A program can be viewed as a *function* that takes input data values, process the values and returns results of the process as the the output.

  - Output = *Program(Input)*
    - ***Example*** z = AddNumbers.sum(x, y); here x and y are input, while z is the output.

Program Example:

```
import java.util.Scanner;
class AddNumbers
{
  public static void main(String args[])
  {
    int x, y, z;
    System.out.println("Enter two integers to
calculate their sum");
    Scanner in = new Scanner(System.in);
    x = in.nextInt();
    y = in.nextInt();
    z = sum(x, y);
    System.out.println("Sum of the integers = " + z);
  }
  public static int sum (int a, int b)
  {
            return a+b;
  }
}
```
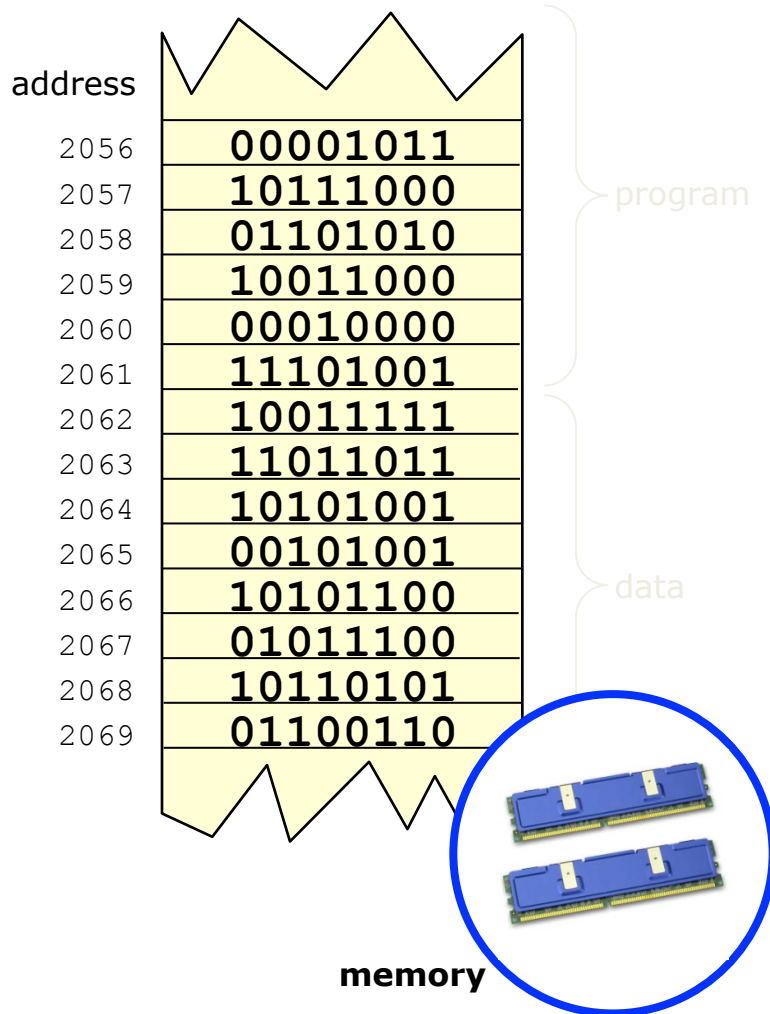
What is a (Computer) Program?

- A computer *program*, also known as *software*, is simply a *set of instructions* to the computer.

- You tell a computer what to do through programs. Without programs, a computer is an empty machine. Computers do not understand human languages, so you need to use computer languages to communicate with them.

- Programs are written using some **programming language**.

What is a computer program...

...from the computers perspectiv?

address	
2056	**00001011**
2057	**10111000**
2058	**01101010**
2059	**10011000**
2060	**00010000**
2061	**11101001**
2062	**10011111**
2063	**11011011**
2064	**10101001**
2065	**00101001**
2066	**10101100**
2067	**01011100**
2068	**10110101**
2069	**01100110**

program

data

memory

- As described earlier, computers use a language of **zeros** and **ones** because digital devices have two stable states, zero and one.

- The instructions (program) given to computer are expressed as a combinations **of ones and zeros**

- Thus, from a computer's perspective, it can be said that **a computer program is a collection of ones and zeros, stored in a memory.**

Linnæus University

What is a computer program...

...from the programmer's perspective?

A computer program is code written in a high level language, that **humans understand**. The high level language is then translated to machine code, which consists of ones and zeros.

A **programmer** formulates or uses an already defined *algorithm* to give instruction to a computer to solve a problem by writing a **computer program**.

An **algorithm** is simply a sequence of operations for performing a specific task.

An Exempel Java Program

```
class AddNumbers
{
    public static void main(String args[])
    {
        int sum;
        sum =37 + 15;
        System.out.println("Sum of the integers 15 and 37 is = " + sum);
    }
}
```

> The computer program sums the digits 37 and 15, and then prints the sum.

Output

Sum of the integers 15 and 37 is = 52

Programming Language

- Programming language, as the name indicates, is simply a language used to express computer programs. Java, C++, C#, Python, COBOL, FOTRAN, etc., all are the examples of programming language.

- Programming languages are classified into three main categories:
    1. **Machine Language**
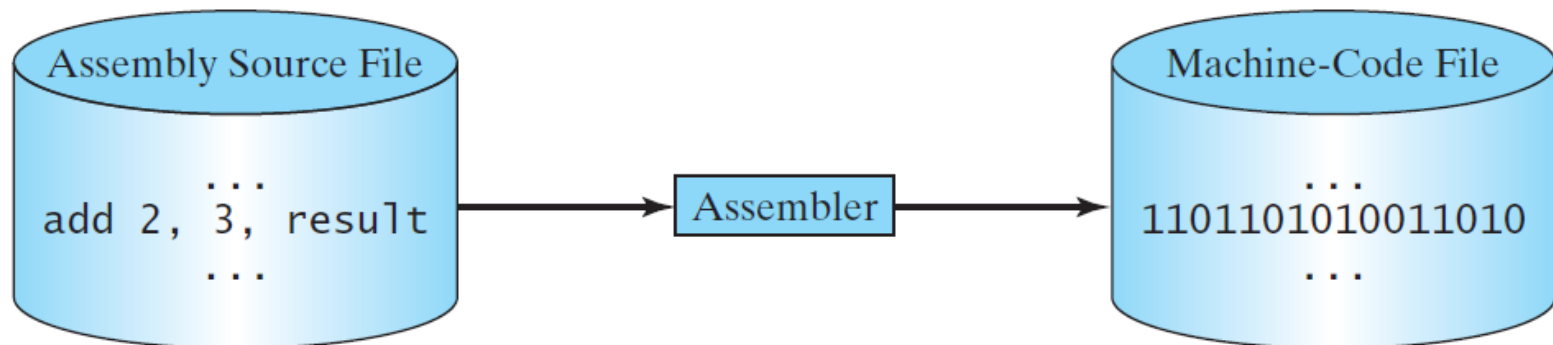   2. **Assembly Language**
    3. **High-Level Language**

Machine Language

- Machine Language is a language *directly understandable by processing unit* of a computer.
- It is a set of primitive instructions built into every computer.
  - The instructions are in the form of **binary code** (0s and 1s), so you have to enter binary codes for various instructions.

- A **problem** with Machine language is that it is *highly difficult to program, read and modify for humans*. For example, to add two numbers, you might write an instruction in binary like this: `11011010100011010`
- To make programming process simple and easier for humans, we need a language that is close to human-language.

Linnæus University

Assembly Language

- **Assembly language** was developed to make programming simple and easier for humans.

- It corresponds closely to the instruction set of a given computer and allows symbolic naming of operations and addresses using English-like codes called *mnemonics*, which are easier to remember and understand as compare to pure binary code.

- A computer cannot understand assembly language, however, a program called **assembler** is used to convert assembly language programs into machine code.

  - For example, to add two numbers, you might write an instruction in assembly code like this: ADDF3 R1, R2, R3

Assembly Source File
```
...
add 2, 3, result
...
```
→ Assembler →
Machine-Code File
```
...
1101101010011010
...
```

High-Level Language

- Although assembly language is an improvement over machine code from a programmer's perspective, it is still tedious to use and work with for humans.

- High-Level Languages, such as Java, C++, C#, etc., were introduced to make writing, reading, and understanding programs even more easier for humans.

- The high-level languages are English-like and easy to learn and program.

- For example, the following is a high-level language statement that computes the area of a circle with radius 5:
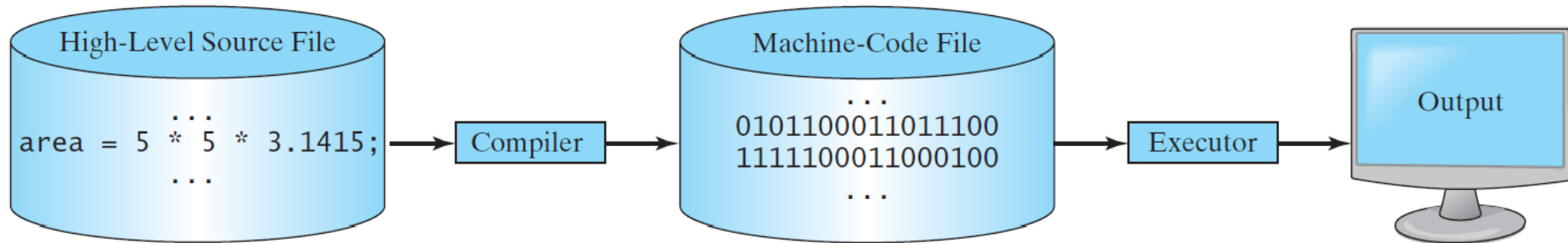
    area = 5 * 5 * 3.1415;

Linnæus University

Compilers and Interpreters

- A program written in a high-level language is called a ***source program*** **or** ***source code***.

- Because a computer cannot understand a source program, a source program must be translated into machine code for execution.

- The translation can be done using either a ***compiler*** *or an interpreter*.
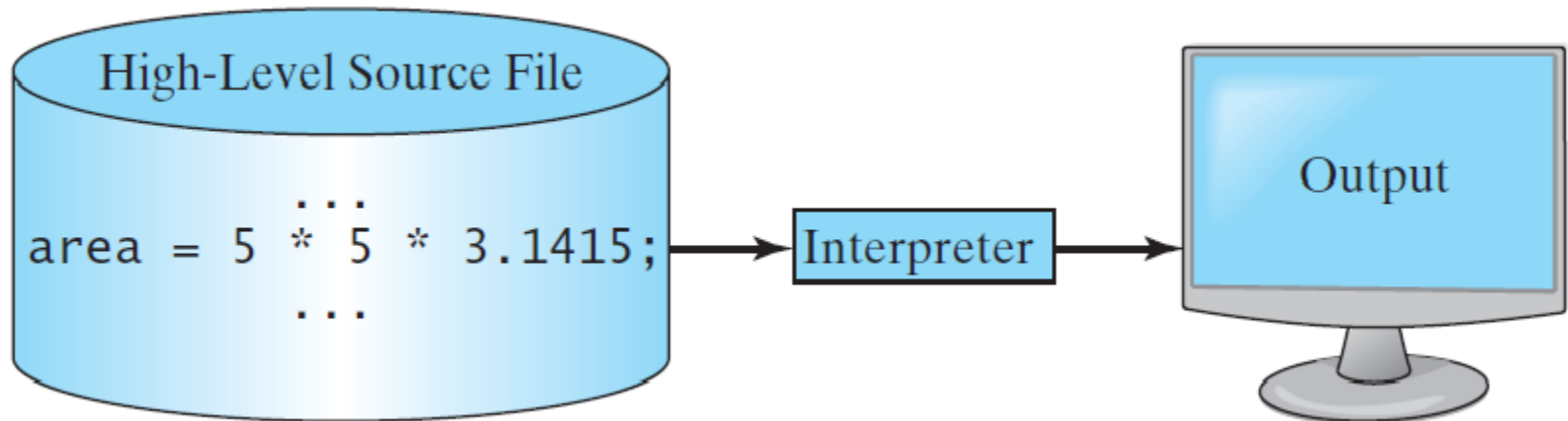
Linnæus University

Compiler

A **compiler** is a special program that translates the *entire* source code into a machine-code file, and the machine-code file is then executed, as shown in the following figure.

High-Level Source File
```
...
area = 5 * 5 * 3.1415;
...
```

Compiler →

Machine-Code File
```
...
0101100011011100
1111100011000100
...
```

Executor →

Output

Linnæus University

Interpreter

An **interpreter** reads **one statement** from the source code, translates it to the machine code or virtual machine code, and then executes it right away, as shown in the following figure.
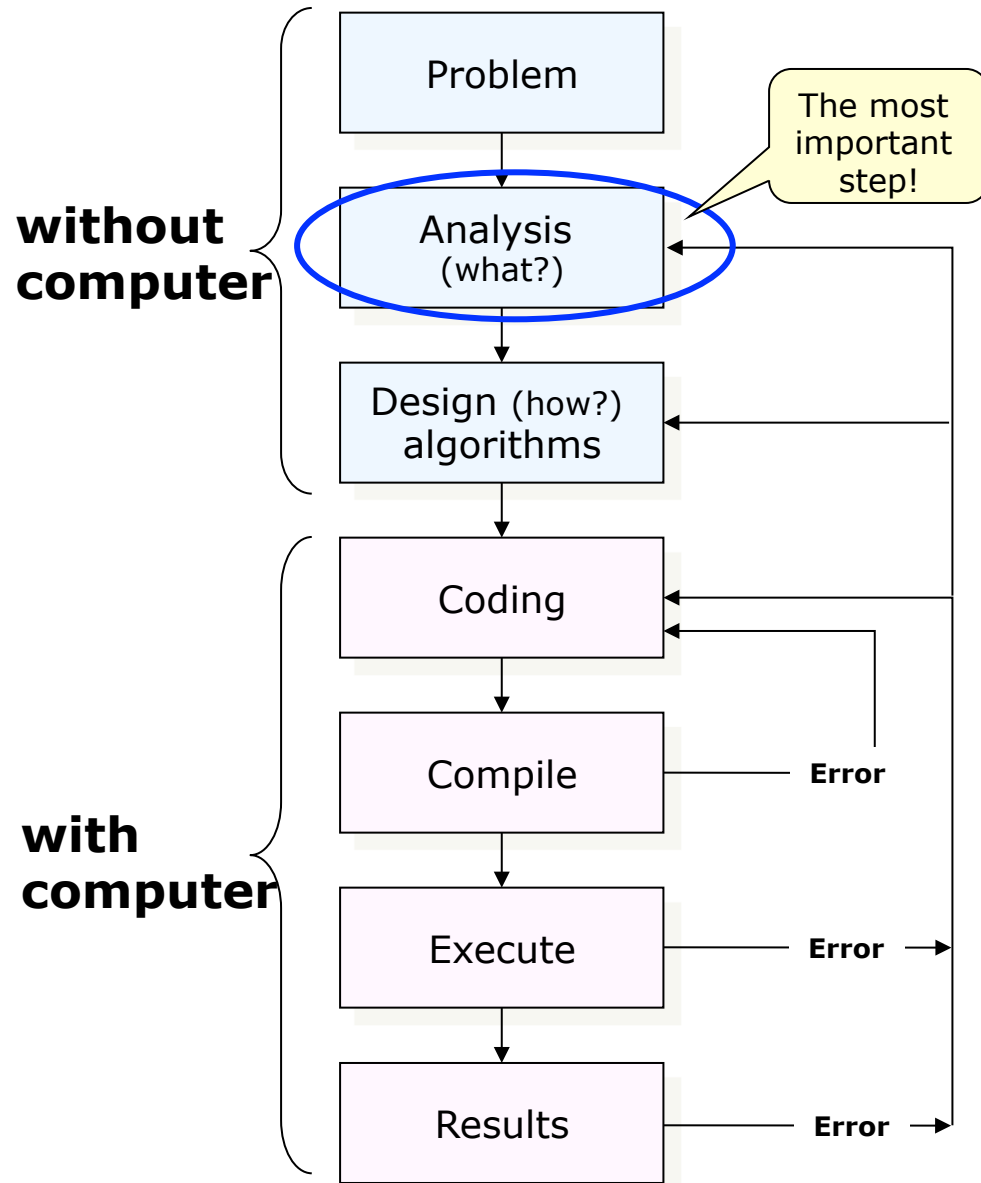
Note that a statement from the source code may be translated into several machine instructions.

Problem Solving
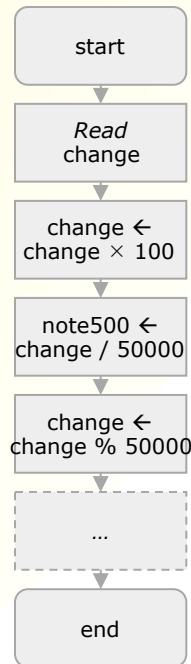
Problem Solving

- The basic purpose of writing a program is to **solve a problem**.

- General steps in problem solving are:

  - **Understanding the problem**.

  - Divide the problem into more manageable smaller problems.

  - Model or create a Solution.

  - Consider alternative solutions.

  - Implement the solution.

  - Test the solution and fix any problems found

without computer

with computer

Problem

Analysis (what?)

The most important step!

Design (how?) algorithms

Coding

Compile — **Error**

Execute — **Error**

Results — **Error**

Linnæus University

Algorithm

Convert change from euros to cents
Calculate the number of 500-bills.
Calculate the remaining change.
Calculate the number of 200-bills.
Calculate the remaining change.
Calculate the number of 100-bills.
Calculate the remaining change.
Calculate the number of 50-bills.
Calculate the remaining change.
Calculate the number of 20-bills
Calculate the remaining change.
Calculate the number of 10-bills.
Calculate the remaining change.
Calculate the number of 5-bills.
Calculate the remaining change.
Calculate the number of 1-coins
Calculate the number of 50-cents
Calculate the remaining change

"in plain English"

- An algorithm is a sequence of instructions that are made in order to accomplish something or to solve a problem.
  - Can be written in plain English, flow chart, pseudo code, ...
  - Might need adjustments as you work.

start

Read
change

change ←
change × 100

note500 ←
change / 50000

change ←
change % 50000

...

end

flow chart

Always construct an algorithm before you start coding!

To understand and to analyse a problem

- **The problem**

  - Convert an exchange in euro to the number of 500-, 100-, 50-, 20-, 10-, 5-bills, 1- euro coins and 50-cent coins.

- **Examples**

  - We have 628,50 euro, i.e. 62850 cents.

  - The number of 500-bills: divide 62850 with 50000

    - The quotient is 1.

    - The remainder is 12850.

  - The number of 100-bills: divide 12850 with 10000

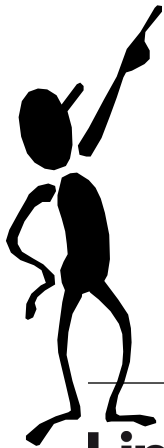   - The quotient is 1.

    - The remainder is 2850

  - Calculate the remaining number of bills and coins using the same principle.

  - Results: 1 pc of 500-bills, 0 pc of 200-bills, 1 pc of 100-bills, 0 pc of 50-bills, 1 pc of 20-bills, 0 pc of 10-bills, 1 pc of 5-bills, 3 pcs of 1-coins and 1 pc of 50-cent coins.
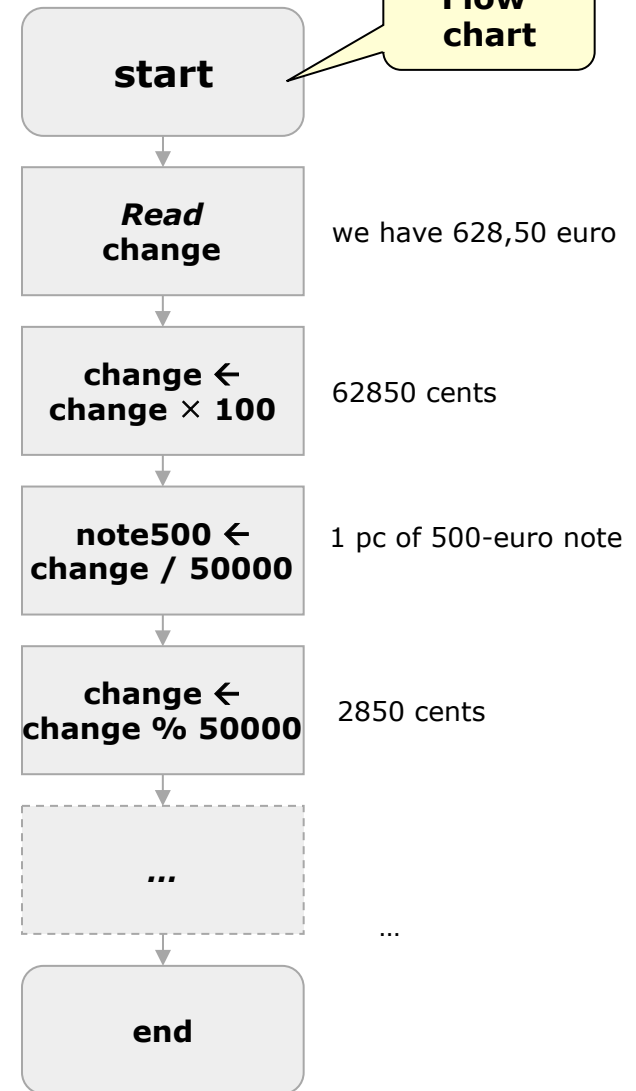
The resulting algorithm

1. Convert change from euros to cents
2. Calculate number of 500-bills. → 62850 cents
3. Calculate the remaining change. → 1 pc 500-bills
4. Calculate number of 200-bills. → 12850 cents
5. Calculate the remaining change. → 0 pc 200-bills
6. Calculate number of 100-bills. → 2850 cents
7. Calculate the remaining change → 1 pc 100-bills
8. Calculate number of 50-bills. → 2850 cents
9. Calculate the remaining change. → 0 pc 50-bills
10. Calculate number of 20-bills. → 2850 cents
11. Calculate the remaining change. → 1 pc 20-bills
12. Calculate number of 10-bills. → 850 cents
13. Calculate the remaining change. → 0 pc 10-bills
14. Calculate number of 5-bills. → 850 cents
15. Calculate the remaining change. → 1 pc 5-bills
16. Calculate number of 1-coins. → 350 cents
17. Calculate the remaining change. → 3 pc 1-coins
18. Calculate number of 50-cents. → 50 cents
    → 1 pc 50-cents

In plain English

A step by step instruction – a sequence!

Flow chart

```
      start
        │
        ▼
   Read change          we have 628,50 euro
        │
        ▼
   change ←             62850 cents
   change × 100
        │
        ▼
   note500 ←            1 pc of 500-euro note
   change / 50000
        │
        ▼
   change ←             2850 cents
   change % 50000
        │
        ▼
      ...               ...
        │
        ▼
      end
```

Implementation

- The final *implementation* of an algorithm means that we translate the developed algorithm into a working program in a programming language that we have chosen.

- We do this using the available *syntax*, i.e., the instructions provided by the programming language, Java for instance, for performing the operations in resp. sequence step.

- The program is then compiled and executed to solve the problem addressed by the program.

Summary

- **A computer** is an electronic machine that can store and process data.

- A computer require instructions for all its operations.

- The instructions are given to computer by using a programming language.

- Programming language is a language used to write computer programs.

- A program is simply a set of instructions.

- The basic purpose of writing a program is to solve a problem.

Suggested Readings

- Introduction to Java Programming, Brief Version, Global Edition, 11/E Liang, Chapter 1