CSI31 Introduction to Computer Programming I

Dr. Sharon Persinger
Fall 2018

Overview

- Basic definitions:
  - Computer
  - Computer science
  - Algorithm
  - Programming language

What is a computer?

- A modern computer is “a machine that stores and manipulates information under the control of a changeable program.
- Store and manipulate information
- Changeable program
- Universal problem solver

What is a program?

- A computer program is a detailed step-by-step set of instructions to a computer.

- A program solves a specific problem.

What is Computer Science?

- "Computer science is no more about computers than astronomy is about telescopes." – Edsger W. Dijkstra

- Computer science is the scientific study of solving problems by computers.

- What problems can we solve by computers? What can be computed?

Related terms and fields

- Mathematics
- Computer programming
- Software engineering
- Many, many subfields

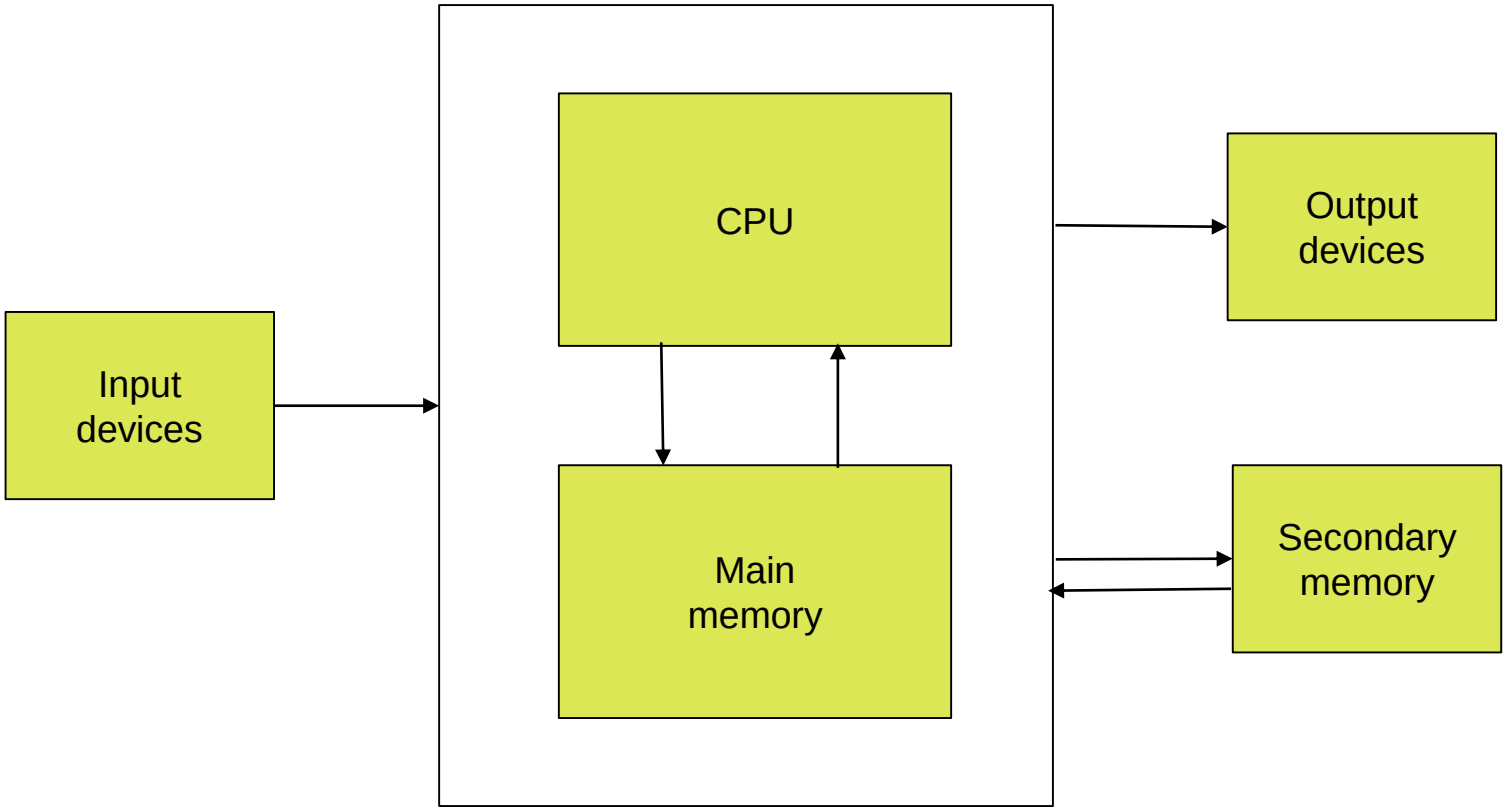
How do you investigate solving a problem?

- Describe the problem clearly and completely. Do you know what a solution is?
- Find a procedure that solves the problem.
- Show that there is a procedure that solves the problem.
- Show that there is no procedure that solves the problem.

Algorithms, computability, complexity

- An algorithm is a step-by-step process for solving a problem.
- Design an algorithm.
  - Understand the problem.
  - Can this problem be solved at all by an algorithm?
- Analyze an algorithm.
  - Is this solution a good one?
  - Is there a better one?
  - What does better mean? Time? Space?

Abstract View of Basic Computer Hardware



What happens when you run a program?

- Program instructions are copied into main memory.
- CPU executes the program step by step:

  fetch-execute cycle.

  - Get the first instruction
  - Decode what it means
  - Carry it out.
  - Then get the next instruction, and so on.

Hardware Details

- CPU or central processing unit – carries out the basic operations, arithmetic and logical, follow fetch-execute cycle

- Main memory, RAM, random access memory – stores program and data while program is working on data, volatile

- Secondary memory – permanent storage of programs and data, hard disk, flash memory, CD

Hardware Details

- Input devices – get information from user into computer,
  - Examples: keyboard, mouse, microphone

- Output devices – present information from computer to user
  - Examples: monitor, printer, speakers

Programming Languages

- Designed for expressing computer operations – arithmetic and logic – without ambiguity
- Constructs have syntax – formal rules
- Constructs have semantics - meaning

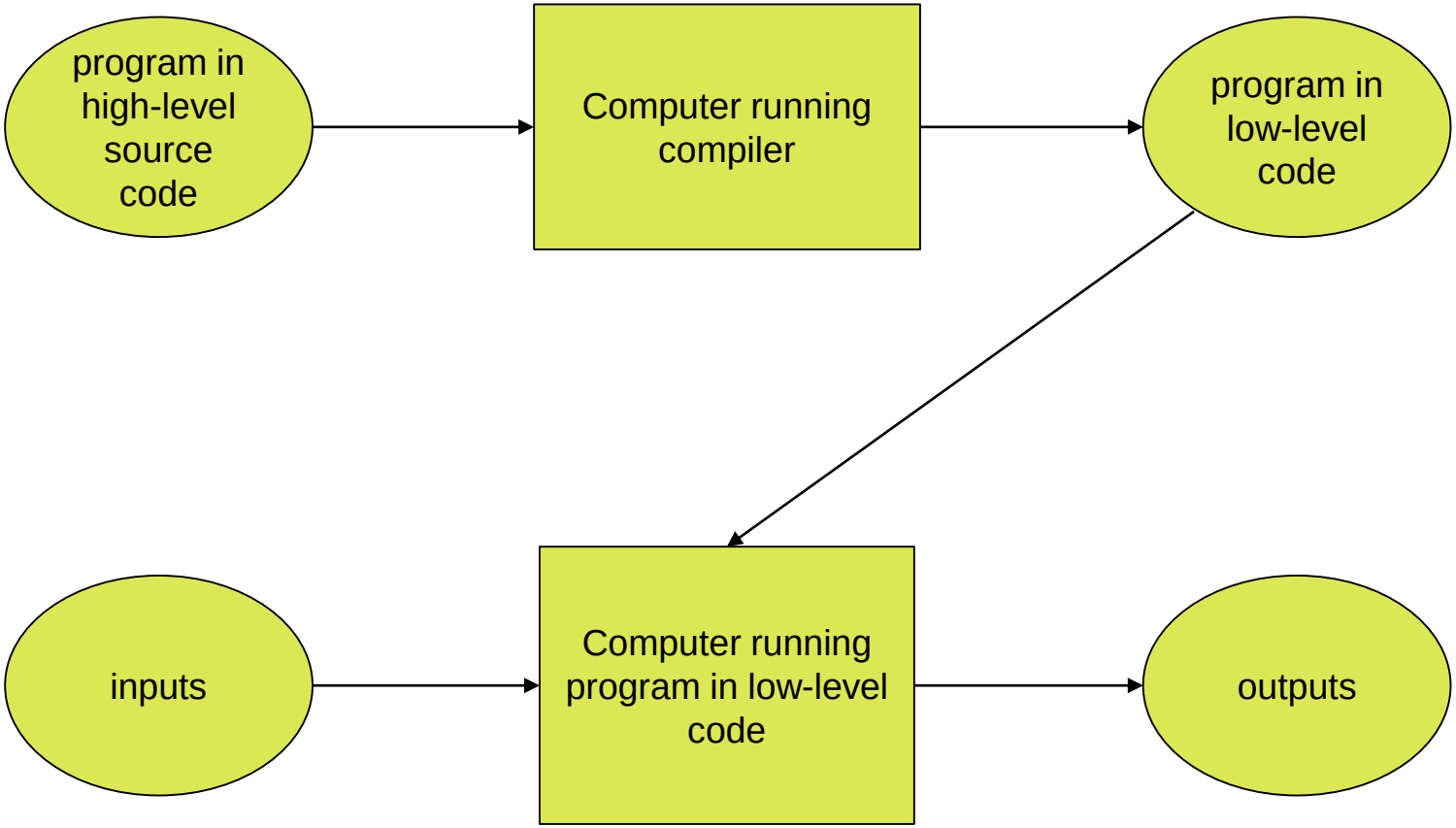
Low Level Language – ASM assembly language

```
main proc
    mov         ax, seg message
    mov         ds, ax
    mov         ah, 09
    lea  dx, message
    int  21h
    mov         ax, 4c00h
    int  21h
main   endp
end main
```

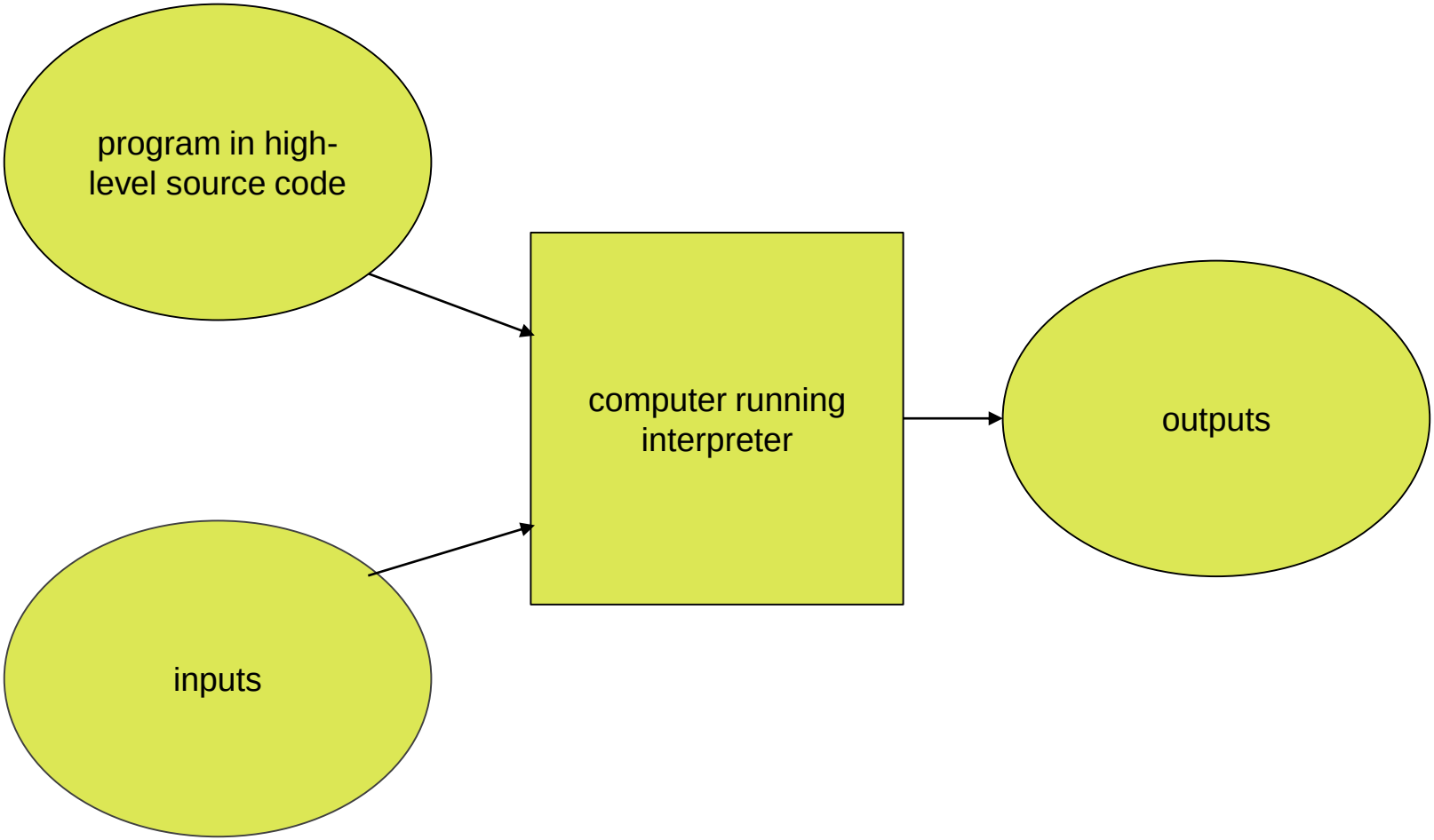
High level vs low level languages

- Low level languages refer to specific memory locations
  - moving numbers from memory location into CPU, computing arithmetic, and moving results back to memory
  - specific to a particular computer architecture
  - for machines to execute
- High level languages use notation like mathematics: a = b+c
  - designed for people to write programs
  - portable to lots of different CPUs

Compiling a program: from high level to low level



Interpreting a program: from high level to low level



Compile vs Interpret

- Compiler translates entire program completely.
  - compiled program can be run over and over without being recompiled
  - compiler not needed to execute code
  - produces faster running programs

- Interpreter analyzes and executes the code instruction by instruction
  - interpreter needed at run time
  - flexible programming

- Python is an interpreted language.

What will you do in this course?

- Learn some of the terminology of computer science: CPU, input device, high-level language, compiler, function, top-down design, …

- Learn to analyze problems and write Python programs that solve those problems.

Before you leave class

- Send an email message with the subject CSI31 to prof.persinger@gmail.com
- Include your name in the body of the message.