

# Welcome to MCS 260

- 1 About the Course and Evaluation
  - catalog, prerequisites, the MCS major
  - goals and expectations
- 2 Computer Science and Algorithms
  - two definitions
  - Python and JupyterLab
  - organization of the content
- 3 Getting Started
  - installing with miniconda
  - every lab session ends with a quiz

MCS 260 Lecture 1  
Introduction to Computer Science  
Jan Verschelde, 12 June 2023

# Introduction to Computer Science

- 1 About the Course and Evaluation
  - catalog, prerequisites, the MCS major
  - goals and expectations
- 2 Computer Science and Algorithms
  - two definitions
  - Python and JupyterLab
  - organization of the content
- 3 Getting Started
  - installing with miniconda
  - every lab session ends with a quiz

# Catalog Description and the MCS Major

## MCS 260. Introduction to Computer Science

Computer literacy, number systems, concepts of operation systems, storage, files, databases, logic gates, circuits, networks, internet.

Introduction to programming in Python, variables, assignments, functions, objects.

**Prerequisite(s):** Credit or concurrent registration in MATH 180.

MCS 260 is the first course in the MCS major.

Mathematical Computer Science (MCS) has two tracks:

- 1 Concentration in Algorithms and Theory;
- 2 Concentration in Computational Mathematics.

MCS 260 *introduces* to computer literacy and programming.

# Introduction to Computer Science

## 1 About the Course and Evaluation

- catalog, prerequisites, the MCS major
- goals and expectations

## 2 Computer Science and Algorithms

- two definitions
- Python and JupyterLab
- organization of the content

## 3 Getting Started

- installing with miniconda
- every lab session ends with a quiz

# Goals and Expectations

Thinking as a computer scientist involves

- 1 the understanding of concepts of computer science; and
- 2 the ability to express algorithms in working code.

Your grade will be determined on a total of 600 points:

- 100 points from quizzes and homework;
- four projects, for a total of 200 points;
- one midterm exam, for 100 points; and
- the final exam, worth 200 points.

By default, unless explicitly stated that collaborations are allowed,  
*all submitted solutions must be your own work.*

Course URL: <http://homepages.math.uic.edu/~jan/mcs260>.

# Introduction to Computer Science

- 1 About the Course and Evaluation
  - catalog, prerequisites, the MCS major
  - goals and expectations
- 2 Computer Science and Algorithms
  - **two definitions**
  - Python and JupyterLab
  - organization of the content
- 3 Getting Started
  - installing with miniconda
  - every lab session ends with a quiz

# Computer Science

Copied from the Association of Computing Machinery (ACM):

## Definition (Computer Science)

*Computer Science* is the systematic study of those algorithms which describe and transform information: the underpinning theory, analysis, planning, efficiency, realization and application.

The ingredients in this definition point at three different fields: (1) mathematics (*theory and analysis*); (2) engineering (*planning, efficiency, and realization*); and (3) business (*application*).

## Definition (Algorithm)

An *algorithm* is an *ordered*<sub>(1)</sub> set of *unambiguous*<sub>(2)</sub>, *executable*<sub>(3)</sub> steps that define a *terminating*<sub>(4)</sub> process.

Source: *Computer Science, an Overview* by J. Glenn Brookshear.

# Introduction to Computer Science

## 1 About the Course and Evaluation

- catalog, prerequisites, the MCS major
- goals and expectations

## 2 Computer Science and Algorithms

- two definitions
- **Python and JupyterLab**
- organization of the content

## 3 Getting Started

- installing with miniconda
- every lab session ends with a quiz



# Python and JupyterLab

Our two main tools are Python and JupyterLab:

- Python is a high level programming language, originally a scripting language for web applications, with computational ecosystems.
- Jupyter stands for Julia, Python, R, and many others. JupyterLab is a web based interactive development environment for notebooks, code, and data.

A notebook documents our work flow.

Python scripts are defined in files with the extension `.py`, while Jupyter notebooks have the extension `.ipynb`.

# Introduction to Computer Science

- 1 About the Course and Evaluation
  - catalog, prerequisites, the MCS major
  - goals and expectations
- 2 Computer Science and Algorithms
  - two definitions
  - Python and JupyterLab
  - organization of the content
- 3 Getting Started
  - installing with miniconda
  - every lab session ends with a quiz

# Organization of the Content

The two parts of the course are separated by the midterm exam.

- 1 For programming “*in the small*”, we cover data structures, control structures, top down functional design.

Computer literacy concepts are organized along

- ▶ hardware: *How does a computer work?* and
- ▶ software: *How to use a computer?*

- 2 For programming “*in the large*” we cover modules, classes and objects, concepts of software engineering.

Intellectual property, software licensing, computer ethics, etc.

By the end of the course, we will prototype a basic video game.

# Introduction to Computer Science

## 1 About the Course and Evaluation

- catalog, prerequisites, the MCS major
- goals and expectations

## 2 Computer Science and Algorithms

- two definitions
- Python and JupyterLab
- organization of the content

## 3 Getting Started

- installing with miniconda
- every lab session ends with a quiz

# installing with miniconda

Conda is a package manager for Python.

- 1 Visit `https://docs.conda.io` and follow the instructions to install `miniconda`.

The `miniconda` is faster to install than the full `anaconda`.

- 2 Type `conda install python` in a conda shell.

- 3 Type `conda install jupyterlab` in a conda shell.

To check, type `python --version` and `jupyter --version`,  
to launch JupyterLab type `jupyter lab` in a conda shell.

# Introduction to Computer Science

## 1 About the Course and Evaluation

- catalog, prerequisites, the MCS major
- goals and expectations

## 2 Computer Science and Algorithms

- two definitions
- Python and JupyterLab
- organization of the content

## 3 Getting Started

- installing with miniconda
- every lab session ends with a quiz

# Exercises

In the lab session of this week, consider the following problems:

- 1 Explain the differences between formulas and algorithms. What are the similarities?
- 2 What is the difference between a heuristic and an algorithm? Illustrate the difference with an example.
- 3 Describe (*give pseudocode for*) the algorithm to compute  $a + b$ , with  $a$  and  $b$  given as sequences of digits.
- 4 Do the previous exercise for the multiplication  $a \star b$ .
- 5 Draw (*give a flowchart for*) the algorithm to compute  $a + b$ , with  $a$  and  $b$  given as sequences of digits.
- 6 Do the previous exercise for the multiplication  $a \star b$ .

A selection of these homework problems will be collected and graded.