CS65: Introduction to Computer Science

September 9, 2025

Syllabus
Logistics
Introduction to Algorithms

Drake
UNIVERSITY

Road Map

- Brief introduction
- Course logistics
- Topics
  - Algorithms
  - Programming
  - Computer program
  - How to program
  - Pseudocode
  - Writing program in a specific language eg Python
  - Integrated Development Environment (IDE)

Introduction

- Dr Md **Alimoor Reza**
  - Assistant Professor of Computer Science, Dept. of Mathematics and Computer Science, Drake University, USA

  - **Office:** Shuang-Zi Building (11th Floor)
  - **Email:** md.reza@drake.edu
  - **Office hours:** Tuesday: 8:00-9:50 AM
    Thursday: 2:00-3:50 PM
    additionally by appointment
    Zoom link

Prior Teaching Experience

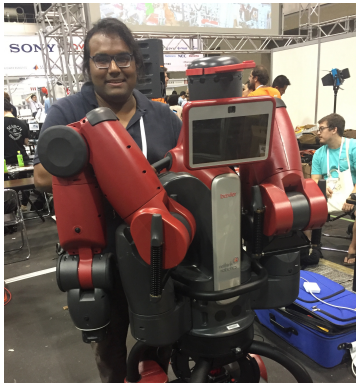
- Prior teaching experience
  - Drake University
    - Introduction to Computer Science (Fall'21&22, Spring'22&23)
    - Computer Vision (Spring'22)
    - Robotics (Fall'22)
    - Artificial Intelligence (Spring'23)
  - Indiana University Bloomington
    - Computer Vision (Spring'21)
    - Artificial Intelligence (Fall'18)
  - George Mason University
    - Introduction to Computing , Object Oriented Programming, Data Mining

About My Research

- Research
  - Passionate about Artificial Intelligence (A.I.) for Robots
  - Studying various types of images and finding meaningful concepts from them

Object Detection

Semantic Segmentation



Introduction

- Additional help
  - Dr. Wei Wang (Associate Professor, Qingdao University)

- Now your turn, briefly introduce yourselves!
  - Your name
  - Hobby/interest

Road Map

- Brief introduction
- Course logistics
- Topics
  - Algorithms
  - Programming
  - Computer program
  - How to program
  - Pseudocode
  - Writing program in a specific language eg Python
  - Integrated Development Environment (IDE)

Course Logistics

- This is an introductory course. Four years of high school mathematics or MATH 20. Logical thinking and basic familiarity with computer and its applications are helpful

- Course contents: lecture slides, assignments, labs can be found on course website:
  - https://analytics.drake.edu/~reza/teaching/cs65_fall25/cs65_schedule_sec1.html
  - https://analytics.drake.edu/~reza/teaching/cs65_fall25/cs65_schedule_sec2.html
  - https://analytics.drake.edu/~reza/teaching/cs65_fall25/cs65_schedule_sec3.html

- I will also upload the lectures, assignments, labs on Fanya
  - https://qddx.mh.chaoxing.com/

Course Logistics

- Coursework for will include:
  - *attending the class (in-person)*
  - *individual and collaborative programming assignments*
  - *lab assignments*,
  - *Content quizzes*
  - *final project*

- Read the syllabus @ Fanya + my course schedule

Course Schedule (tentative)

analytics.drake.edu/~reza/teaching/cs65_fall25/cs65_schedule_sec1.html

CS 65: Introduction to Computer Science (Fall 25)
Instructor: **Dr. Md Alimoor Reza**
Assistant Professor of Computer Science
Department of Mathematics and Computer Science
Drake University

Classroom: Boyi#508@Qingdao University
Meeting Time : Tuesday 14:00 pm - 15:50 pm (Week#1-Week#9), Thursday 10:10 am - 12:00 pm (Week#1-Week#17)
Office Hours: Tuesday: 08:00 am-9:50 am and and Thursday: 2:00 pm - 3:50 pm or by appointment

MOOC Portal: Chaoxing Fanya (Xue Xi Tong)

Section#1 Schedule
A tentative schedule is provided below (content may be adjusted as we progress).

Date	Topic	Reading	Items due
week 1 (Tue: Sep 09)	Introduction to Computer Science Lecture 1 slide **Lab 0 (Windows user)** **Lab 0 (Mac user)**	Reading: A Byte of Python (Why Python & its advantage)	
week 1 (Thu: Sep 11)	Variables, expression, and statements Lecture 2 slide **Lab 1 (released on Sep 11)**	Reading:	

Course Schedule (tentative)

analytics.drake.edu/~reza/teaching/cs65_fall25/cs65_schedule_sec2.html

CS 65: Introduction to Computer Science (Fall 25)
Instructor: **Dr. Md Alimoor Reza**
Assistant Professor of Computer Science
Department of Mathematics and Computer Science
Drake University

Classroom: Boyi#508@Qingdao University
Meeting Time : Tuesday 16:00 pm - 17:50 pm (Week#1-Week#17), Friday 14:00 pm - 15:50 pm (Week#1-Week#9)
Office Hours: Tuesday: 08:00 am-9:50 am and and Thursday: 2:00 pm - 3:50 pm or by appointment

MOOC Portal: Chaoxing Fanya (Xue Xi Tong)

Section#2 Schedule
A tentative schedule is provided below (content may be adjusted as we progress).

Date	Topic	Reading	Items due
week 1 (Tue: Sep 09)	Introduction to Computer Science Lecture 1 slide **Lab 0 (Windows user)** **Lab 0 (Mac user)**	Reading: A Byte of Python (Why Python & its advantage)	
week 1 (Fri: Sep 12)	Variables, expression, and statements Lecture 2 slide **Lab 1 (released on Sep 12)**	Reading:	

Course Schedule (tentative)

analytics.drake.edu/~reza/teaching/cs65_fall25/cs65_schedule_sec3.html

CS 65: Introduction to Computer Science (Fall 25)
Instructor: **Dr. Md Alimoor Reza**
Assistant Professor of Computer Science
Department of Mathematics and Computer Science
Drake University

Classroom: Boyi#508@Qingdao University
Meeting Time : Wednesday 14:00 pm - 15:50 pm (Week#1-Week#9), Friday 08:00 am - 09:50 pm (Week#1-Week#17)
Office Hours: Tuesday: 08:00 am-9:50 am and and Thursday: 2:00 pm - 3:50 pm or by appointment

MOOC Portal: Chaoxing Fanya (Xue Xi Tong)

Section#3 Schedule
A tentative schedule is provided below (content may be adjusted as we progress).

Date	Topic	Reading	Items due
week 1 (Wed: Sep 10)	Introduction to Computer Science Lecture 1 slide **Lab 0 (Windows user)** **Lab 0 (Mac user)**	Reading: A Byte of Python (Why Python & its advantage)	
week 1 (Fri: Sep 11)	Variables, expression, and statements Lecture 2 slide **Lab 1 (released on Sep 12)**	Reading:	

Course Expectations

There is no programming prerequisite for this course

> As long as you can type and turn on a computer (and think logically), you can earn a letter grade “A” in this course

Those with programming experience … be careful

Office Hours: Shuang-Zi Building 11th Floor)

Day	Time
Tuesday	8 am - 9:50 am
Thursday	2 pm - 3:50 pm

The above times I guarantee I will be available in my office (Shuang-Zi Building 11th Floor)

- If you have a question, swing on by!
- In practice, I'm usually responsive to questions from students via email between 8:00am and 8:00pm

Textbook

It's been my experience that most students can do well in the course using the lecture notes and code examples supplied

I don't want to make you purchase a text that will go out of date or that you won't use again

There are plenty of free, open-source, online textbooks

A small inconvenience: there isn't one online textbook that does it all

I don't love *everything* about any text

I will post links to chapters of a textbook that will provide supplementary materials for you for each class

Textbook

A Byte of Python by Swaroop

Think Python: How to Think Like a Computer Scientist 2nd Edition by Allen B. Downey

A Practical Introduction to Python Programming by Brian Heinold

Python for Everybody by Charles Severance

Here are what people are saying about the textbook

A Byte of Python by Swaroop

> First of all, I want to say thanks to you for this great book. I think it is a good book for those who are looking for a beginner's tutorial for Python. It is about two or there years ago, I think, when I first heard of this book. At that time, I was unable to read books in English yet, so I got a chinese translation, which took me into the gate of Python programming. Recently, I reread this book. This time, of course, the english version. I couldn't believe that I can read the whole book without my dictionary at hand. Of course, it all dues to your effort to make this book an easy-to-understand one. -- myd7349

Reference: https://github.com/swaroopch/byte-of-python

Text reading before next class

Read *About Python* https://python.swaroopch.com/about_python.html from “A Byte of Python”

Things to look for:

- Why is it called Python?
- Advantages of Python

Grading Policy

- *Homework Assignments (20%).*
  - homework Python programming activities (3-4 in total).
- *Programming Labs (35%).*
  - A series of short exercises accompanying nearly each class topic. Relatively easier than homework assignments (10-12 in total).
- *Content Quizzes (30%).*
  - "paper-and-pencil exams" quizzes based on the lecture contents. (4-6 in total)
- *Final project (10%).*
  - Individual proposed project. You also need to prepare a powerpoint presentation by the end of semester.
- *Attendance (5%).*
  - Counted based on your signature.

Grading Policy

Programming Labs are a series of short exercises accompanying nearly each class topic, usually culminating in a challenge exercise which will be submitted for credit

- Submitted via Fanya (to be described on next class)
- There will be many of these (likely around 10-12)
- 4 points each

Grading Policy

Homework Assignments are more challenging Python programming challenges than Python. You also need to submit for credit

- Submitted via Fanya (to be described on next class)
- There will be likely around 3-4 homework assignments

Grading Policy

Content Quizzes are open-note, open-book quizzes that will be submitted via blackboard

Their *tentative* dates will be assigned and released soon on

Grading Policy

Final Project

You will develop a program of your choosing that incorporates many different programming concepts you have learned throughout the semester.

The final project is to be completed individually and is due at the end of the semester.

You also need to prepare an accompanying PowerPoint presentation.

Grading Policy

Attendance and Participation Policy

The most important thing is to **show up** (or let me know if you aren't able to)

Why?

- You learn more
- It helps me adjust the pace of the course

Each class will contain some activity that will require your participation and will be recorded on the Fanya website.

Attendance and Participation Policy

If for any reason, you are unable to make the in-person class session, please email me.

If you are unable to participate in the course for a prolonged period, we will discuss whether an incomplete is the best option.

If you have 5 or more unexcused absences, you will automatically receive an F for the course.

Attendance and Participation Policy

I plan on learning all of your names, so please be patient while I navigate through the course roster

Additionally, I will also maintain your attendance for every class by having you add your signature in a sheet of paper. By the end of the semester, I will count all the signatures from those attendance sheets to calculate your attendance score.

Grading Policy

Grading scale: The tentative grading scale for this course would be as follows discretion):

- A+ (97%-100%)
- A (91%-96%)
- A- (83%-90%)
- B+ (77%-82%)
- B (73%-76%)
- B- (69%-72%)
- C+ (66%-68%)
- C (63%-65%)
- C- (62%)
- D+/D (61%)
- D- (60%)
- F (55%)

Can I use ChatGPT or other AI Assistance to help?

Short answer: yes. But, cite it

- A brief statement in your assignment indicating what assistance was provided by ChatGPT.
- Filling out the “AI Assisted Learning Reflection” questionnaire. Link provided in the blackboard website.

Writing computer code is like handwriting

If you use AI to write code for you, and don’t cite it, it’s fairly obvious to me

You can not use ChatGPT on the content quizzes (but you can use anything accessible via Fanya such as readings or your own code)

Collaboration Policy and Academic Integrity

- You may work together on ***labs***
- You may **not** work together on projects and quizzes
- When working together, say who you worked with in the code comments
- When taking code from somewhere on the Internet, cite it in your comments and provide a link
- May not look at code from previous students in the course
- May not search the Internet for solutions
- First violation: 0 on that thing
- Second violation: F for the course

Summary: Course Logistics

- Please read the syllabus on Fanya prior to next class

- Important highlights:
  - Grading requirement and grading scale
  - Attendance policy
  - Deadline policy
  - Technology requirement
  - Collaboration and Academic integrity

Introductory Questionnaire

Before next class, please answer the questions on the introductory Questionnaire on Fanya

This take-home questionnaire is intended:

- to help the professor learn more about you, and
- to help us both understand your initial thoughts about this class, including any goals and expectations you have for the upcoming semester.

Do you have any questions?

So far…. before next class

1. Read syllabus
2. Read a chapter from online text

Road Map

- Brief introduction
- Course logistics
- Topics
  - Algorithms
  - Programming
  - Computer program
  - How to program
  - Pseudocode
  - Writing program in a specific language eg Python
  - Integrated Development Environment (IDE)

Introduction to Computer Science

Study of Computers?

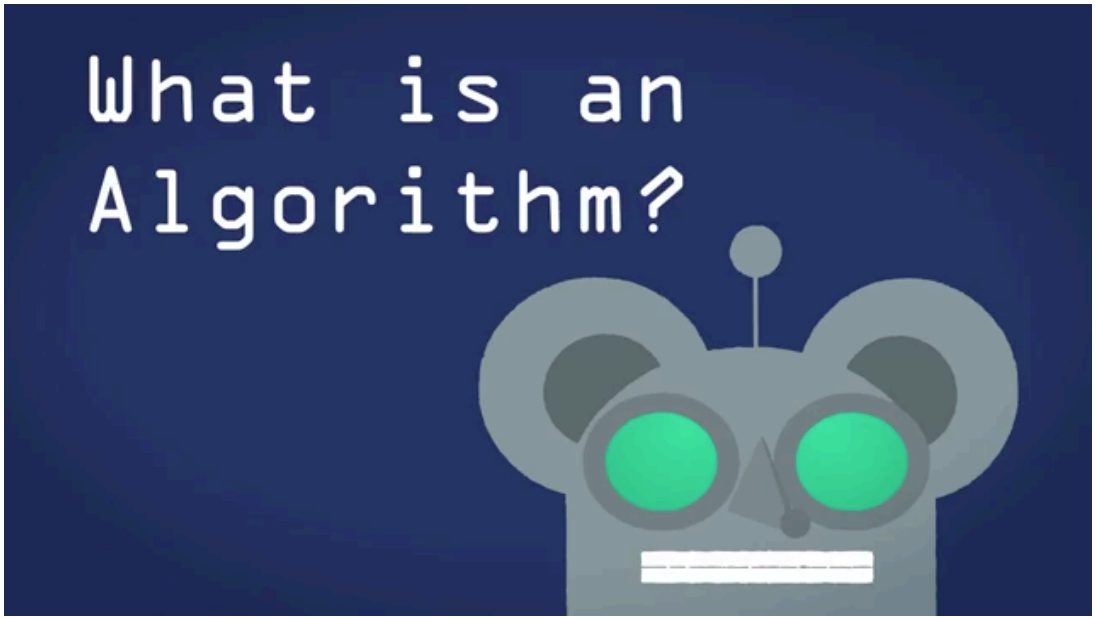
Programming?

Computer science is the study of **Algorithms**

Algorithm: A step-by-step procedure for producing a solution to a given problem

Introduction to Computer Science

- Focus of this course would be the study of computer algorithms
- Algorithm
  - A crucial component in problem solving
  - step-by-step instructions to be executed by the machine
  - here is a toy example

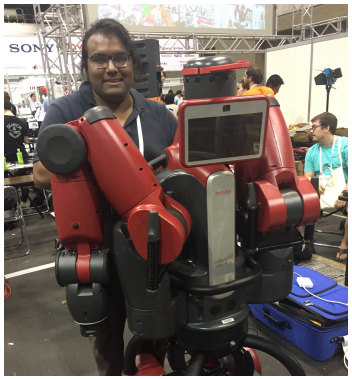


Reference: BBC Learning

Introduction to Computer Science

- Algorithm:
  - step-by-step instructions to be executed by the machine
  - more realistic algorithm in robotics (from my research)

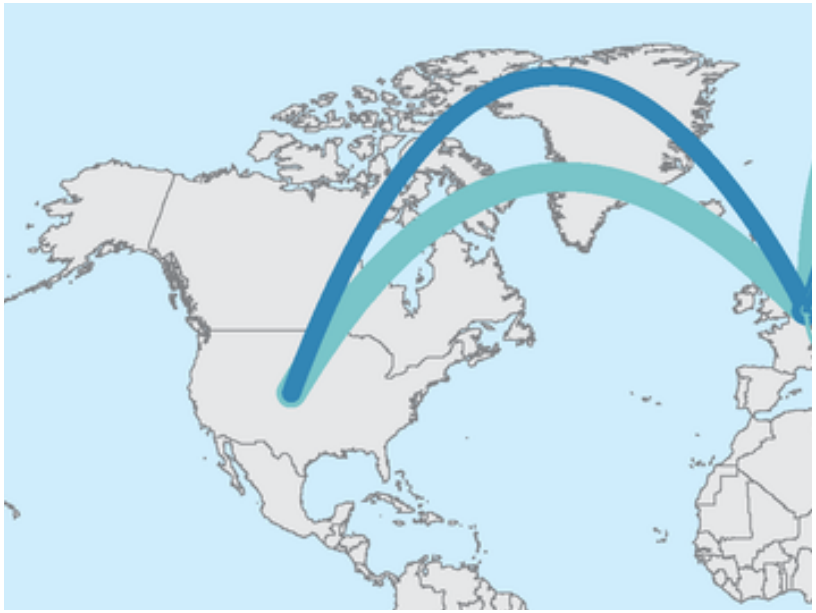
Baxter robot with a goal of fetching an specific object ie, Pringles from the shelf



Me with Baxter

Introduction to Computer Science

- **Algorithm:**
  - step-by-step instructions to be executed by the machine
  - your turn! let's do a group activity for 5 minutes
  - Describe the process of making a trip from USA to Europe?
    - Person next to you is your partner
    - Write down the steps



What is programming and where can I use a program?

- Computers/computing devices are everywhere
  - desktop, laptop, smartphone, robot, microwave, etc



- Computers/computing devices are tools that can be programmed to perform many functions:
  - implement algorithms to solve problems
  - calculate numbers
  - watch videos
  - write notes
  - design models etc

What are computer programs?

- A program is a sequence of instructions that specifies how to perform a computation
  - can be written by a specific programming language

- Programming languages are formal language to express computations
  - **Python**
  - Java
  - C/C++

- Programming languages have strict rules, known as syntax that must be followed
  - Specific keywords need to be used to perform some action
  - Specific structure to be followed
  - Naming convention

How to write a program?

- Programs must be designed before they are written

- Program development cycle
  - Design a program (eg, in pseudocode)
  - Write the code (in a programming language like python)
  - Correct the syntax errors
  - Test the program
  - Correct the logic errors

Pseudocode

- Pseudocode is an informal language that has no syntax
  - the examples we have seen before (brushing tooth, robot grasping, making Euro trip) can be written in pseudocode

```
1.. If student's grade is greater than or equal to 60
        Print "passed"
else
        Print "failed"
```

example pseudocode

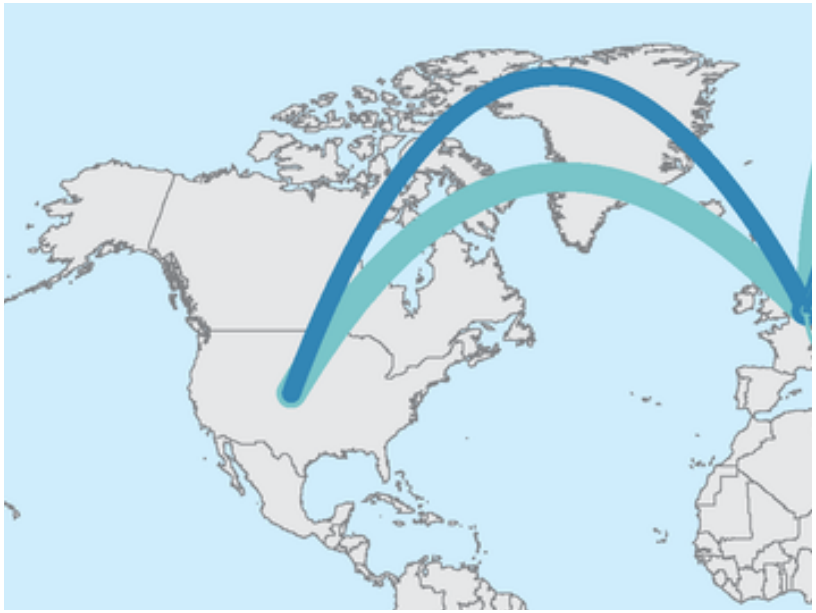
- Not meant to be compiled or executed

- Used to create model program
  - less worry about the syntax, more focus on the logic/design
  - Can be translated into actual code in any programming language eg, **Python**

- I encourage you to always start with pseudocode before you build a program.

In-class Exercise

- Describe the process of making a trip from USA to Europe?
  - Person next to you is your partner
  - Now write down your previous solution in pseudocode



Python

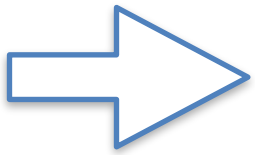
- Python is a programming language

- Purpose is to convert
  - what a programmer writes —> machine executable instructions
- Programmer writes python source code following specific syntax
- There is an interpreter (another program which executes computer code)

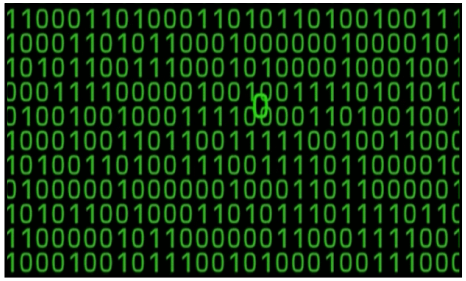
```
n = 5
string = "Hello!"

print(string * n)
```

Python code



Interpreter



Machine readable instructions

Python Keywords

- Programming languages have keywords/reserved words
  - Words that have specified meaning as part of the syntax of the language

- Python's keywords

```
and       del       from      not       while
as        elif      global    or        with
assert    else      if        pass      yield
break     except    import    print
class     exec      in        raise
continue  finally   is        return
def       for       lambda    try
```

Integrated Development Environment

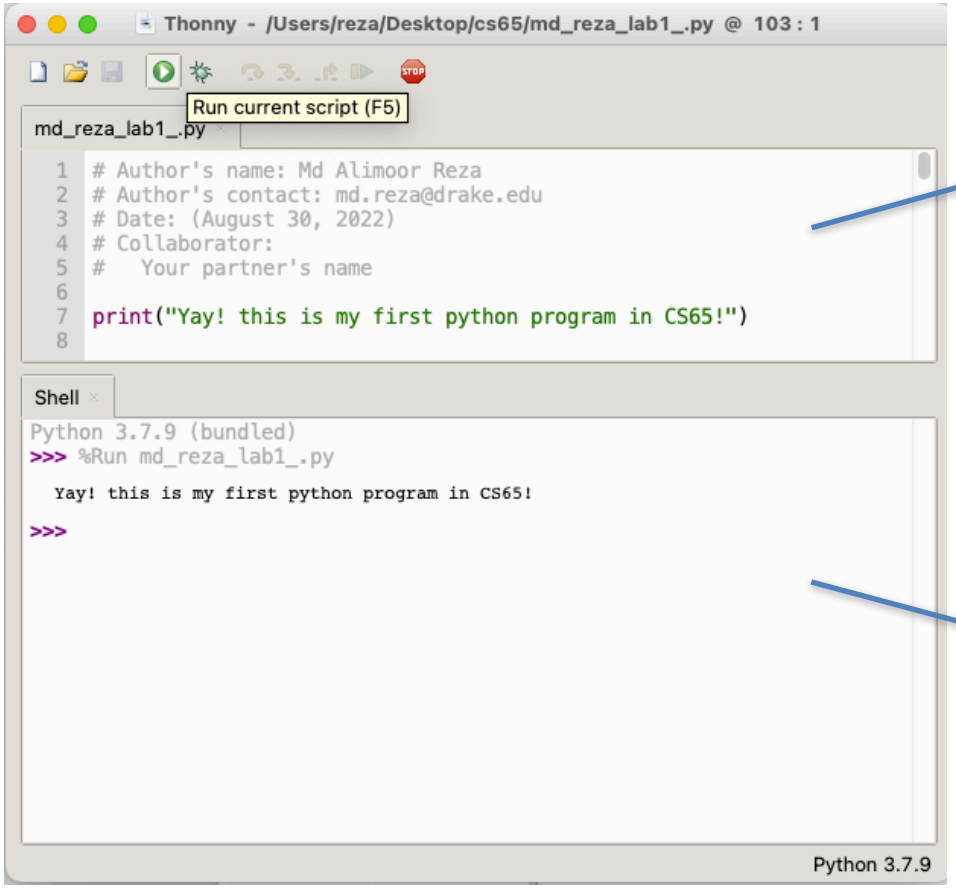
- Integrated Development Environment (IDE) is tool or software system that programmers use to create, run, and test new programs
  - text editor
    - writing python code
  - compiler/interpreter
    - for translating the code into machine understandable instructions
  - executable environment
    - for showing the result of the program

- We will be using the Python programming language along with an IDE for creating Python programs

- Thonny as an IDE
  - very user friendly tool
  - freely available online

Integrated Development Environment

- Demo by Reza in his computer



Summary

- Main takeaway from this lecture:
  - We will be studying algorithms
  - You will be implementing various algorithms using Python programming language in Thonny (IDE)

- I prepared a document for your if you want to install Thonny on your personal laptop
  - Will be released on my course website as well as on Fanya

Summary

- To do: Read syllabus

- To do: Read a chapter from online text

analytics.drake.edu/~reza/teaching/cs65_fall25/cs65_schedule_sec1.html

CS 65: Introduction to Computer Science (Fall 25)
Instructor: **Dr. Md Alimoor Reza**
Assistant Professor of Computer Science
Department of Mathematics and Computer Science
Drake University

Classroom: Boyi#508@Qingdao University
Meeting Time : Tuesday 14:00 pm - 15:50 pm (Week#1-Week#9), Thursday 10:10 am - 12:00 pm (Week#1-Week#17)
Office Hours: Tuesday: 08:00 am-9:50 am and and Thursday: 2:00 pm - 3:50 pm or by appointment

MOOC Portal: Chaoxing Fanya (Xue Xi Tong)

Section#1 Schedule
A tentative schedule is provided below (content may be adjusted as we progress).

Date	Topic	Reading	Items due
week 1 (Tue: Sep 09)	Introduction to Computer Science Lecture 1 slide **Lab 0 (Windows user)** **Lab 0 (Mac user)**	Reading: A Byte of Python (Why Python & its advantage)	
week 1 (Thu: Sep 11)	Variables, expression, and statements Lecture 2 slide **Lab 1 (released on Sep 11)**	Reading:	