

# EEE110 - Computer Programming (Python)

## Week 1: Course Introduction and Scope



**ADANA ALPARSLAN TÜRKES  
SCIENCE AND TECHNOLOGY UNIVERSITY**

Dr Kasım Zor

Department of Electrical and Electronic Engineering

Spring 2022

# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



# Course Instructor

Dr Kasım Zor

Electrical and Electronic Engineer, PhD

## Research Interests

- Electrical Energy and Power Systems, Electric Load Forecasting, Energy Analytics, Artificial Intelligence, and Renewable Energy

## Contact Information

- Office: Room 210, West Wing, Prefab Building
- E-mail: [kzor@atu.edu.tr](mailto:kzor@atu.edu.tr)
- Web: [www.kasimzor.com.tr](http://www.kasimzor.com.tr)



# Laboratory Assistant

Ms Elanur Ekici, Research Assistant  
Electrical and Electronic Engineer, PhD Candidate

## Research Interests

- Power Electronics

## Contact Information

- Office: Room 127, North Wing, Prefab Building
- E-mail: [EEKICI@ATU.EDU.TR](mailto:EEKICI@ATU.EDU.TR)



# Course Information

Course Title	Code	Semester	T+L (Hours)	Credits	ECTS
Computer Programming	EEE110	2	3+2	4	6

**Table 1:** Table of Course Information

- Prerequisites: None
- Level: Bachelor
- Language: English
- Type: Compulsory



# Course Assessment and Evaluation

Assessment Type	Quantity	Weight
Midterm Examination	1	40%
Final Examination	1	60%

**Table 2:** Table of Course Assessment and Evaluation

	Course Type	Allowed Rate	Allowed Hours
Absentee Rate	Main Course	30%	13
	Laboratory	20%	6

**Table 3:** Table of Absentee Rate



# Laboratory Schedule

---

## Lab Contents

---

W1 Introduction to the Laboratory

W2 Introduction to Anaconda

W3 Introduction to Python Programming Language

W4 Decision Structures & Boolean Logic

W5 Repetition Structures

W6 Functions

W7 Files and Exceptions & Lists and Tuples

W8 Strings & Dictionaries and Sets

---

W10 Classes and Object-Oriented Programming

W11 Inheritance, Polymorphism, and Recursion

W12 Array-Oriented Programming with Numerical Python (NumPy)

W13 Introduction to Python Data Analysis (Pandas)

W14 GUI Programming (Tkinter)

W15 Database Programming

---





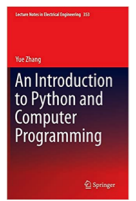
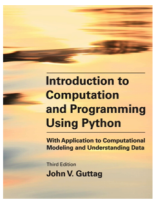
# Learning Outcomes

- Understand Python basics and use it for procedural, array-oriented, object-oriented, and GUI programming
- Able to manipulate a variety of Python data types
- Able to detect and fix common errors in Python programs
- Able to write small-scale computer programs via Python

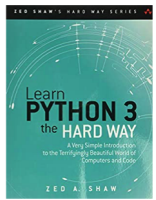
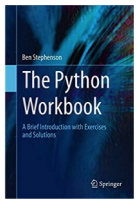


# Recommended Sources

## Textbooks [1, 2, 3, 4]



## Additional Resources [5, 6, 7, 8, 9]



# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



## Course Contents – Week 2

### Introduction to Computers and Programming

Rank	Language	Web <sup>1</sup>	Mobile <sup>2</sup>	Enterprise <sup>3</sup>	Embedded <sup>4</sup>	Score
1	Python	X		X	X	100.0
2	Java	X	X	X		95.4
3	C		X	X	X	94.7
4	C++		X	X	X	92.4
5	Javascript	X				88.1
6	C#	X	X	X	X	82.4
7	R			X		81.7
8	Go	X		X		77.7
9	HTML	X				75.4
10	Swift		X	X		70.4

<sup>1</sup>Web: Languages used for developing web sites and applications

<sup>2</sup>Mobile: Languages used for applications on mobile devices

<sup>3</sup>Enterprise: Languages used for enterprise, desktop, and scientific applications

<sup>4</sup>Embedded: Languages used to program device controllers

**Table 4:** IEEE Spectrum Top Programming Languages [10]



# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language**
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



# Course Contents – Week 3

## Introduction to Python Programming Language

- JupyterLab,
- Jupyter Notebook,
- Spyder.

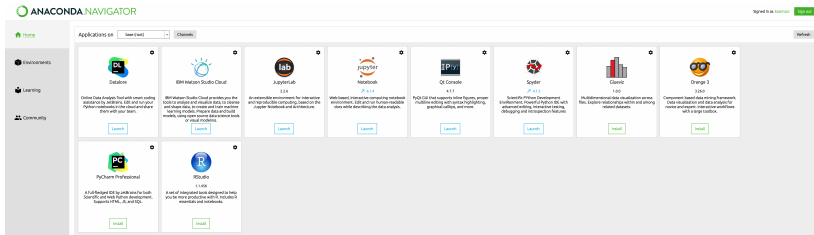


Figure 1: Anaconda Distribution for Python



# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic**
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



# Course Contents – Week 4

## Decision Structures and Boolean Logic

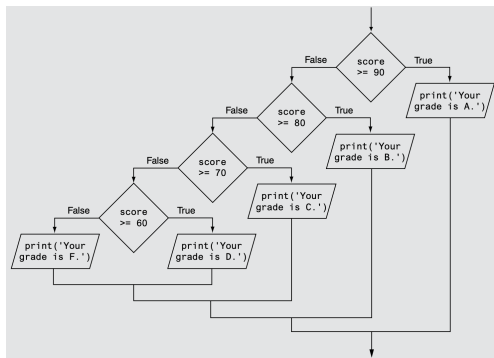


Figure 2: Demonstration of a decision structure example [1]





# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures**
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



# Course Contents – Week 5

## Repetition Structures

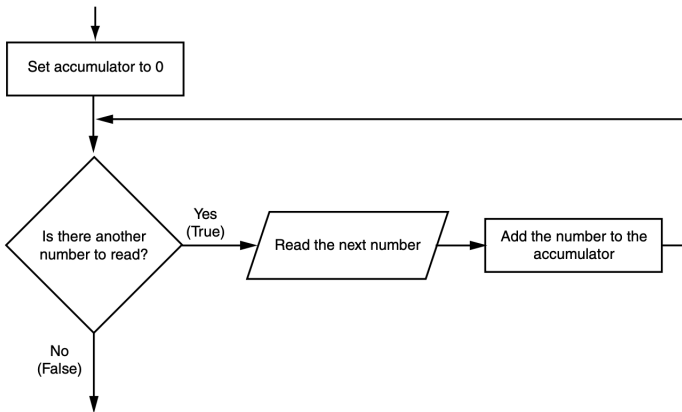


Figure 3: Illustration of a repetition structure example [1]



# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions**
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



# Course Contents – Week 6

## Functions

These statements cause  
 the message function to  
 be created.

```
# This program demonstrates a function.
# First, we define a function named message.
def message():
    print('I an Arthur,')
    print('King of the Britons.')
```

```
# Call the message function.
message()
```

This statement calls  
 the message function,  
 causing it to execute.

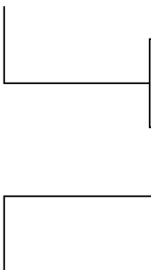


Figure 4: Demonstration of a function example [1]



# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting**
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



# Course Contents – Week 7

## Files and Exceptions, Lists and Tuples, & Introduction to Plotting

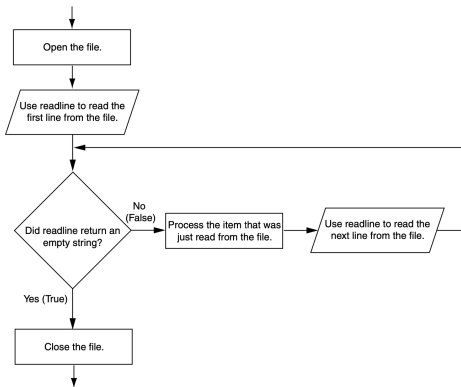


Figure 5: Flowchart of a file process [1]



# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets**
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



## Course Contents – Week 8

### Strings & Dictionaries and Sets

```

>>> name = 'Kelly' Enter # name is 'Kelly'
>>> name += ' ' Enter # name is 'Kelly '
>>> name += 'Yvonne' Enter # name is 'Kelly Yvonne'
>>> name += ' ' Enter # name is 'Kelly Yvonne '
>>> name += 'Smith' Enter # name is 'Kelly Yvonne Smith'
>>> print(name) Enter
Kelly Yvonne Smith
>>>

```

Figure 6: An example of string concatenation [1]





# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination**
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



## Course Contents – Week 9

### Midterm Examination (Paper-Based)

#	Difficulty	Minutes	Pts	Scope
Q1	Very Easy	5	10	W1–W3
Q2	Easy	10	20	W3–W5
Q3	Moderate	30	30	W6–W8
Q4	Hard	45	40	W6–W8
Total		90	100	W1–W8

**Table 5:** An Example of Midterm Examination Assessment



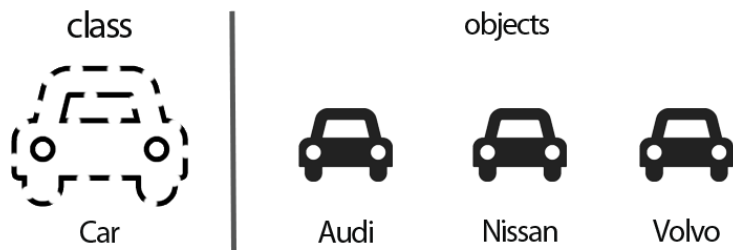
# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming**
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



# Course Contents – Week 10

## Classes and Object-Oriented Programming



**Figure 7:** Demonstration of objects and classes [11]



# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion**
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



# Course Contents – Week 11

## Inheritance, Polymorphism, and Recursion [1]

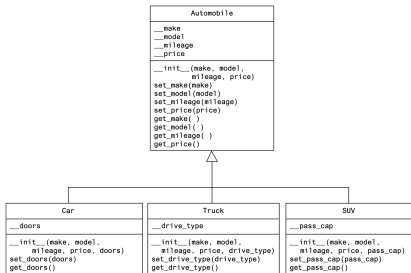


Figure 8: Inheritance

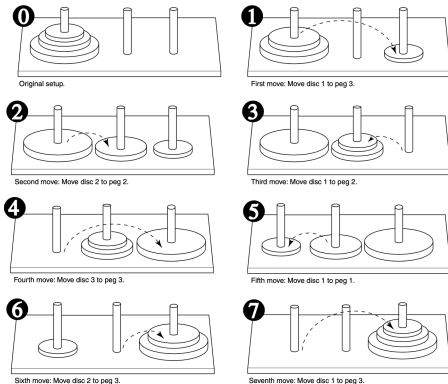


Figure 9: Recursion



# Outline

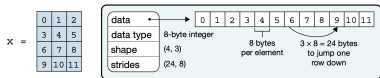
- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy**
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



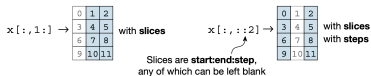
# Course Contents – Week 12

## Array-Oriented Programming with NumPy

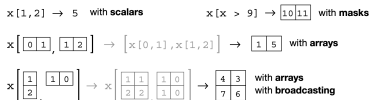
### a Data structure



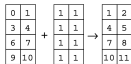
### b Indexing (view)



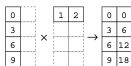
### c Indexing (copy)



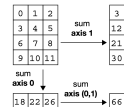
### d Vectorization



### e Broadcasting



### f Reduction



### g Example

```
In [1]: import numpy as np
In [2]: x = np.arange(12)
In [3]: x = x.reshape(4, 3)

In [4]: x
Out [4]:
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11]])

In [5]: np.mean(x, axis=0)
Out [5]: array([4.5, 5.5, 6.5])

In [6]: x = x - np.mean(x, axis=0)

In [7]: x
Out [7]:
array([[ -4.5, -4.5, -4.5],
       [-1.5, -1.5, -1.5],
       [ 1.5,  1.5,  1.5],
       [ 4.5,  4.5,  4.5]])
```

Figure 10: Several fundamental array concepts [12]





# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)**
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination



# Course Contents – Week 13

## Introduction to Python Data Analysis (Pandas) [13]



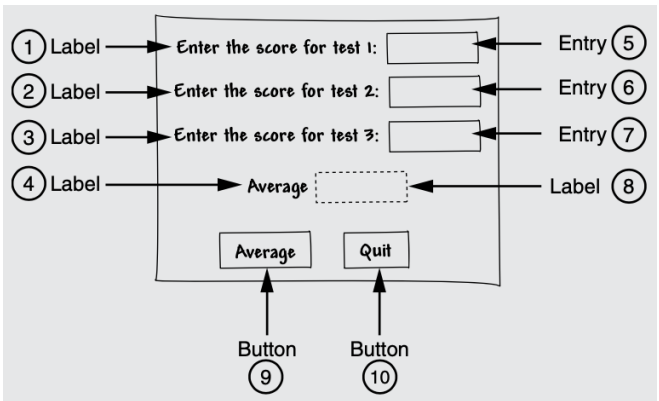
# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming**
- 15 Database Programming
- 16 Final Examination



# Course Contents – Week 14

## GUI Programming [1]



# Outline

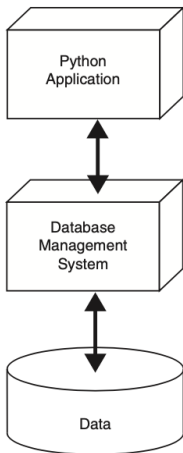
- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming**
- 16 Final Examination



# Course Contents – Week 15 [1]

## A Python application interacting with a DBMS, which manipulates data

---



# Outline

- 1 Course Introduction and Scope
- 2 Introduction to Computers and Programming
- 3 Introduction to Python Programming Language
- 4 Decision Structures and Boolean Logic
- 5 Repetition Structures
- 6 Functions
- 7 Files and Exceptions, Lists and Tuples, & Introduction to Plotting
- 8 Strings & Dictionaries and Sets
- 9 Midterm Examination
- 10 Classes and Object-Oriented Programming
- 11 Inheritance, Polymorphism, and Recursion
- 12 Array-Oriented Programming with NumPy
- 13 Introduction to Python Data Analysis (Pandas)
- 14 GUI Programming
- 15 Database Programming
- 16 Final Examination**



# Course Contents – Week 16

## Final Examination (Paper-Based)

#	Difficulty	Minutes	Pts	Scope
Q1	Very Easy	5	10	W10–W15
Q2	Easy	10	20	W10–W15
Q3	Moderate	30	30	W11–W14
Q4	Hard	45	40	W11–W12
Total		90	100	W1–W15

**Table 6:** An Example of Final Examination Assessment





# References I

- [1] Tony Gaddis. *Starting Out with Python*. Pearson, 5th edition, 2022. ISBN 978-1-292-40863-7.
- [2] John V. Guttag. *Introduction to Computation and Programming Using Python with Application to Computational Modeling and Understanding Data*. MIT Press, 3rd edition, 2021. ISBN 978-0-262-54236-4.
- [3] Paul Deitel and Harvey Deitel. *Intro to Python for Computer Science and Data Science: Learning to Program with AI, Big Data, and the Cloud*. Pearson Education, 1st edition, 2022. ISBN 978-1-292-36490-2.
- [4] Yue Zhang. *An Introduction to Python and Computer Programming, Lecture Notes in Electrical Engineering*, volume 353. Springer, 2015. ISBN 978-981-287-608-9.
- [5] Allan B. Downey. *Modeling and Simulation in Python*. 2017.
- [6] Ben Stephenson. *The Python Workbook A Brief Introduction with Exercises and Solutions*. Springer, 1st edition, 2014. ISBN 978-3-319-14239-5.



## References II

- [7] Zed A. Shaw. *Learn Python 3 the Hard Way A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code*. Addison-Wesley, 1st edition, 2017. ISBN 978-0-13-469288-3.
- [8] Al Sweigart. *Automate the Boring Stuff with Python Practical Programming for Total Beginners*. No Starch Press, 2nd edition, 2020. ISBN 978-1-59327-992-9.
- [9] Kenneth Reitz and Tanya Schlusser. *The Hitchhiker's Guide to Python Best Practices for Development*. O'Reilly, 1st edition, 2016. ISBN 978-1-491-93317-6.
- [10] IEEE Spectrum. The top programming languages 2021, 24<sup>th</sup> Aug, 2021. URL <https://spectrum.ieee.org/top-programming-languages-2021>.
- [11] Java Tutorial Network. What is oop?, 2020. URL <https://javatutorial.net/wp-content/uploads/2014/11/class-object-featured-image.png>.



## References III

- [12] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with numpy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2. URL <https://doi.org/10.1038/s41586-020-2649-2>.
- [13] commbox. How data, analysis, and reports can improve customer service, 2022. URL <https://www.commbbox.io/how-data-analysis-and-reports-can-improve-customer-service/>.

