

Learning Guide Unit 1

Site: [University of the People](#)
Course: CS 1101-01 Programming Fundamentals - AY2024-T1
Book: Learning Guide Unit 1

Printed by: Hoa Doan
Date: Friday, 15 September 2023, 7:32 PM

Table of contents

Overview

Class Introduction

Introduction

Reading Assignment

Discussion Assignment

Programming Assignment

Self-Quiz

Checklist

Overview

Unit 1: Introduction and Fundamental Concepts

Topics:

- Introduction to fundamental programming concepts
- Computer programs
- Debugging
- Errors
- Formal and natural languages

Learning Objectives:

By the end of this Unit, you will be able to:

1. Describe fundamental concepts in computer programming.
2. Explain the difference between high and low-level languages.
3. Demonstrate basic debugging techniques.
4. Reproduce simple calculations in Python.

Tasks:

- Read the Learning Guide and Reading Assignments
- Download and install a working Python environment (not required for PythonAnywhere users). Refer to the Software Requirements/Installation section of the Syllabus.
- Participate in the Discussion Assignment (post and comment in the Discussion Forum)
- Complete and submit the Programming Assignment
- Take the Self-Quiz

Class Introduction

Let's get to know each other! Please share a few things about yourself with the rest of our class, like your name, where are you from, community activities, career aspirations, and so on.

In addition, please answer the question below:

- How do you believe that studying Programming Fundamentals will help you to be more successful in the future? You only need to write what you are comfortable sharing, but please at least say hello to the group.

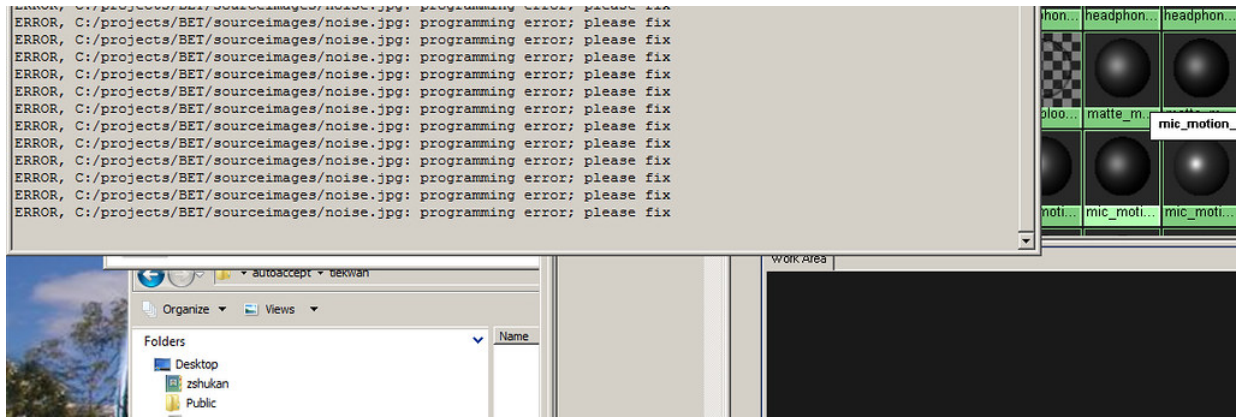
As you can see - there are no rules for what makes a "good" introduction. Instead, your goal is to start class with a smile and find out how you can connect with your classmates. Perhaps someone is from the same area, or you share the same hobby - or have the same type of pet! Respond to a few of your peers' posts and share things you may have in common. **You will find a discussion forum located on the main course page in the General Information and Forums called Class Introductions.**

NOTE: Visit the Class Introductions throughout the first week to respond to fellow classmates' posts and respond to the answers to anyone who responds to your post.

Introduction

Welcome to unit 1!

Unit 1 provides an overview of the problem-solving process, which is a key competency for any computer science professional. As you review the resources in the Reading Assignment, you will learn to formulate problems, think creatively about solutions, and express a solution in a clear and accurate way as you practice solving problems. Additionally, you will gain knowledge on a few crucial topics related to problem-solving, including an introduction to basic programming concepts, computer programs, debugging, errors, and formal and natural languages.



The reading for Unit 1 offers a thorough investigation focusing on:

- foundational concepts in computer programming
- the distinction between high-level languages like Python and low-level languages like assembly language
- basic debugging techniques.

You'll start learning the fundamentals of programming by becoming familiar with the steps needed to install Python on your system as well as how to create and run your first Python program.

Attribution

"programming error, please fix" by zboog is licensed under CC BY-ND 2.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nd/2.0/?ref=openverse>.

Reading Assignment

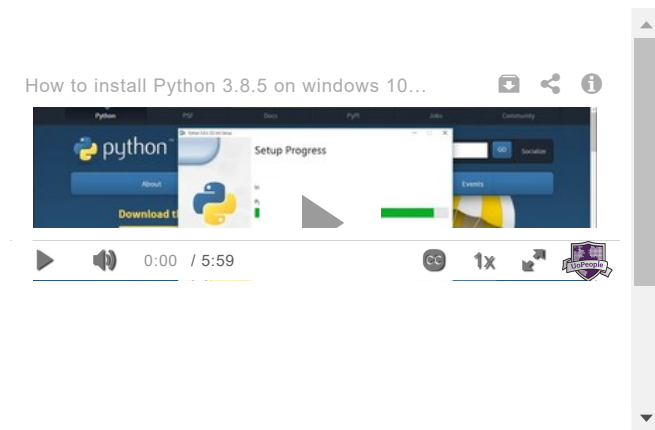
Reading Assignments are designed to expand your knowledge of the topics introduced in the Overview and provide the knowledge you will need to successfully complete the assignments in this unit. As you read through the learning resource and watch the videos consider the following:

- What is the difference between formal and natural languages?
- What are some differences between low-level and high-level programming languages?
- Do you know of any relations between these programming languages and either natural or formal languages?

Read

1. [Think Python: How to think like a computer scientist](#)
Chapter 1 - The Way of the Program

Watch: How to install Python 3.8.5 on Windows 10



References

1. Downey, A. (2015). *Think Python: How to think like a computer scientist*. Green Tea Press.
<https://greenteapress.com/thinkpython2/thinkpython2.pdf>
2. Iqbal, W. (2020, July 27). *How to install Python 3.8.5 on Windows 10 [2020]* - YouTube. [Video].
YouTube. <https://www.youtube.com/watch?v=yiCUmJon-5g>

Discussion Assignment

Hello and welcome to your first computer science assignment! Before you begin exploring the world of programming, you must finish the task of setting up the programming environment. Let's get started!

Download and install a working Python environment, preferably Python 3, or get an account with PythonAnywhere. Please refer to the Software Requirements/Installation section located in the course syllabus for details.

Type the statements below into your Python interpreter. **For each statement**, copy the output into your Discussion Assignment and **explain the output**. Compare it with any similar examples in the textbook, and describe what it means about your version of Python.

```
>>> print 'Hello, World!'
>>> 1/2
>>> type(1/2)
>>> print(01)
>>> 1/(2/3)
```

The code and its output must be explained technically whenever asked. The explanation can be provided before or after the code, or in the form of code comments within the code. For any descriptive type of question, your answer must be at least 150 words.

End your discussion post with one question related to programming fundamentals learned in this unit from which your colleagues can formulate a response or generate further discussion. Remember to post your initial response as early as possible, preferably by Sunday evening, to allow time for you and your classmates to have a discussion.

When you reply to your peers' submissions, compare their results with yours.

Your instructor will grade this assignment using [this rubric](#).

NOTE: Remember to post as early as possible, preferably by Sunday evening, in order to allow time for you and your classmates to discuss.

Programming Assignment

Assignment Instructions

In this unit, we explored the fundamental concepts in computer programming, basic debugging techniques and reproducing simple calculations in Python.

Before completing this assignment, review the reading materials listed below:

a. [Think Python: How to think like a computer scientist](#)

- Review Chapter 1 - The Way of the Program (pp. 1 – 8) and the examples presented in it.

b. For installing the IDLE development environment on UNIX:

- Using Python on Unix platforms. (n.d.). Python Documentation. <https://docs.python.org/3/using/unix.html>

For Windows, refer to:

- Using Python on Windows. (n.d.). Python Documentation. <https://docs.python.org/3/using/windows.html>

For Mac, refer to:

- Savage, B. (n.d.). 5. Using Python on a Mac. Python Documentation <https://docs.python.org/3/using/mac.html>

Part 1:

LEARN FROM YOUR MISTAKES!

In Chapter 1, section 1.9 Exercises (pg. 7) of your textbook, Exercise 1.1 suggests making mistakes when experimenting with a new programming feature. This kind of experiment helps you remember what you read; and to also get information on the error messages.

Answer the following questions (a) to (d).by explaining some frequently occurring errors.

To help you answer the questions, write the code in python and run it to produce output for each of the questions.

- a. If you are trying to print your name, what happens if you leave out one of the quotation marks or both and why?
- b. What is the difference between * and ** operators in Python? Explain with the help of an example.
- c. In Python, is it possible to display an integer like 09? Justify your answer.
- d. Run the commands `type('67')` and `type(67)`. What is the difference in the output and why?

Copy and paste both the code and the screenshot of the output for each of the questions in a Word document.

Part 2:

Write Python program for each of the following questions (a) to (d).

- a. To multiply your age by 2 and display it. For example, if your age is 16, so $16 * 2 = 32$
- b. To Display the name of the city, country and continent you are living in.
- c. To display the examination schedule (i.e., the starting and the ending day) of this term.
- d. Display the temperature of your country on the day the assignment is attempted by you

Add the **Python codes and** screenshots of the **related outputs** in the same word document created for Part 1 submissions. Also **explain what you learned** from the results of each experiment in part 2 questions.

The code and its output must be explained technically. The explanation can be provided before or after the code. The descriptive part of your response must be at least 200 words.

Submission Instructions:

- Submit the solutions to both part 1 and part 2 on the same document.
- Make sure your submission is double-spaced, using Times New Roman, 12-point font, with 1" margins.
- Use sources to support your arguments. Use high-quality, credible, relevant sources to develop ideas that are appropriate for the discipline and genre of the writing.
- Use APA citations and references to support your work. Add a reference list at the end of the submission. For assistance with APA formatting, view the [Learning Resource Center: Academic Writing](#).
- Your submission should be clearly written, concise, and well organized, and free of spelling and grammar errors. The grading will be based on the quality of your analysis, accurate solution of the problem and the quality of your writing.

This assignment will be assessed by your instructor using the rubric available on the assignment page located on the course main page

References

Downey, A. (2015). Think Python: How to think like a computer scientist. Green Tree Press.
<https://greenteapress.com/thinkpython2/thinkpython2.pdf>

Self-Quiz

The Self-Quiz gives you an opportunity to self-assess your knowledge of what you have learned so far.

The results of the Self-Quiz do not count towards your final grade, but the quiz is an important part of the University's learning process and it is expected that you will take it to ensure understanding of the materials presented. Reviewing and analyzing your results will help you perform better on future Graded Quizzes and the Final Exam.

Please access the Self-Quiz on the main course homepage; it will be listed inside the Unit.

Checklist

- Read the course syllabus
- Read the Learning Guide and Reading Assignments
- Introduce yourself (through the Class Introduction section)
- Download and install a working Python 3 environment (not required for Virtual Computer Lab or PythonAnywhere users). Refer to the Software Requirements/Installation section of the Syllabus.
- Participate in the Discussion Assignment (post, comment in the Discussion Forum)
- Complete and submit the Programming Assignment
- Take the Self-Quiz