



CPIT 110

Instructor Manual

For **50** Minutes Lectures

Week 2

8/9/2019 – 12/9/2019

Chapter 1

Introduction to Computers, Programs, and Python

| | |
|-------------------------|-------------------------------|
| This Week Events | – Lab #1 (Chapter 1 – Part 1) |
| Next Week Events | – Lab #2 (Chapter 1 – Part 2) |



CPIT 110

Instructor Manual – Lecture #1 in Week 2

| | |
|--------------------|--|
| Chapter | 1. Introduction to Computers, Programs, and Python |
| Number of Lectures | 3 (50 minutes / Lecture) |
| Lecture | 1 of 3 |
| Slides | 1 - 31 |
| Date | Sunday 08/09/2019 |

| | |
|------------------|--------------------------|
| Week 2 | Lecture 1 of 3 |
| | Slides 1 - 31 |

Topics to Be Covered

❖ 1.2. What is a Computer?

- 1.2.1. Central Processing Unit
- 1.2.2. Bits and Bytes
- 1.2.3. Memory
- 1.2.4. Storage Devices
- 1.2.5. Input and Output Devices

❖ 1.3. Programming Languages

- 1.3.1. Machine Language
- 1.3.2. Assembly Language
- 1.3.3. High-Level Language

❖ 1.4. Operating Systems

- Introduction

Topics Not to Be Covered

- 1.4.1. Controlling and Monitoring System Activities
- 1.4.2. Allocating and Assigning System Resources
- 1.4.3. Scheduling Operations

Learning Objectives

| Learning Outcomes | Topics |
|--|----------------------------|
| – To demonstrate a basic understanding of computer hardware. | 1.2. What is a Computer? |
| – To demonstrate a basic understanding of computer programs. | 1.3. Programming Languages |
| – To demonstrate a basic understanding of operating systems. | 1.4. Operating Systems |

Exercises

❖ 1.2. What is a Computer

1. What are hardware and software?
2. List five major hardware components of a computer.
3. What is a bit? What is a byte?
4. What unit is used to measure disk size?
5. What is the primary difference between memory and a storage device?

❖ 1.3. Programming Languages

1. What language does the CPU understand?
2. What is an assembly language?
3. What is a high-level programming language?
4. What is a source program?
5. What is an interpreter?
6. What is a compiler?
7. What is the difference between an interpreted language and a compiled language?

❖ 1.4. Operating Systems

1. What is an operating system? List some popular operating systems.

CPIT 110

Instructor Manual – Lecture #2 in Week 2

| | |
|--------------------|--|
| Chapter | 1. Introduction to Computers, Programs, and Python |
| Number of Lectures | 3 (50 minutes / Lecture) |
| Lecture | 2 of 3 |
| Slides | 32 - 65 |
| Date | Tuesday 10/09/2019 |

| | |
|-------------------------|--------------------------|
| Week 2 | Lecture 2 of 3 |
| | Slides 32 - 65 |

Topics to Be Covered

- ❖ **1.5. The History of Python**
 - 0.2.1. How Do We Write a Program?
 - 0.2.2. Problem-Solving Phase
 - 0.2.3. Implementation Phase
- ❖ **1.6. Getting Started with Python [...]**
 - 1.6.1. Launching Python
 - 1.6.2 .Creating Python Source Code Files

Learning Objectives

| Learning Outcomes | Topics |
|---|----------------------------------|
| – To describe the history of Python. | 1.5. The History of Python |
| – To explain the basic syntax of a Python program. – To write and run a simple Python program. | 1.6. Getting Started with Python |

Exercises

❖ 1.5 The History of Python

1. Python is interpreted. What does that mean?
2. Can a program that is written in Python 2 runs in Python 3?
3. Can a program written using Python 3 run in Python 2?

❖ 1.6. Getting Started with Python

1. You can run Python in two modes. Explain these two modes.
2. Is Python case sensitive?
3. What is the Python source filename extension by convention?
4. What is a comment? How do you denote a comment line and a comment paragraph?
5. What is the statement to display the message Hello world on the console?
6. Identify and fix the errors in the following code:

```
1 # Display two messages
2 print("Welcome to Python")
3 print("Python is fun")
```

7. Show the output of the following code:

```
1 # Display two messages
2 print("3.5 * 4 / 2 - 2.5 is")
3 print(3.5 * 4 / 2 - 2.5)
```

CPIT 110

Instructor Manual – Lecture #3 in Week 2

| | |
|--------------------|--|
| Chapter | 1. Introduction to Computers, Programs, and Python |
| Number of Lectures | 3 (50 minutes / Lecture) |
| Lecture | 3 of 3 |
| Slides | 66 - 91 |
| Date | Thursday 12/09/2019 |

| | |
|-----------------------------|--------------------------|
| Week 2 | Lecture 3 of 3 |
| | Slides 66 - 91 |

Topics to Be Covered

- ❖ **1.6. Getting Started with Python [...Continued]**
 - 1.6.3. Using Python to Perform Mathematical Computations
- ❖ **1.7. Programming Style and Documentation**
 - 1.7.1 Appropriate Comments and Comment Styles
 - 1.7.2. Proper Spacing
- ❖ **1.8. Programming Errors**
 - Syntax Errors
 - Runtime Errors
 - Logic Errors

Topics Not to Be Covered

- 1.9. Getting Started with Graphics Programming

Learning Objectives

| Learning Outcomes | Topics |
|--|--|
| – To explain the importance of and provide examples of proper programming style and documentation. | 1.7 .Programming Style and Documentation |
| – To explain the differences between syntax errors, runtime errors, and logic errors. | 1.8 .Programming Errors |



Exercises

❖ 1.7. Programming Style and Documentation

1. What is documentation?
2. What is the importance of documentation?
3. How to document code?
4. What is the importance of the proper programming style and documentation?
5. Give a good example of proper programming style and documentation.
6. Give a bad example of proper programming style and documentation.

❖ 1.8. Programming Errors

1. What are three kinds of program errors?
2. If you forget to put a closing quotation mark on a string, what kind of error will be raised?
3. If your program needs to read data from a file, but the file does not exist, an error would occur when running this program. What kind of error is this?
4. Suppose you write a program for computing the perimeter of a rectangle and you mistakenly write your program so that it computes the area of a rectangle. What kind of error is this?