

Chapter 1

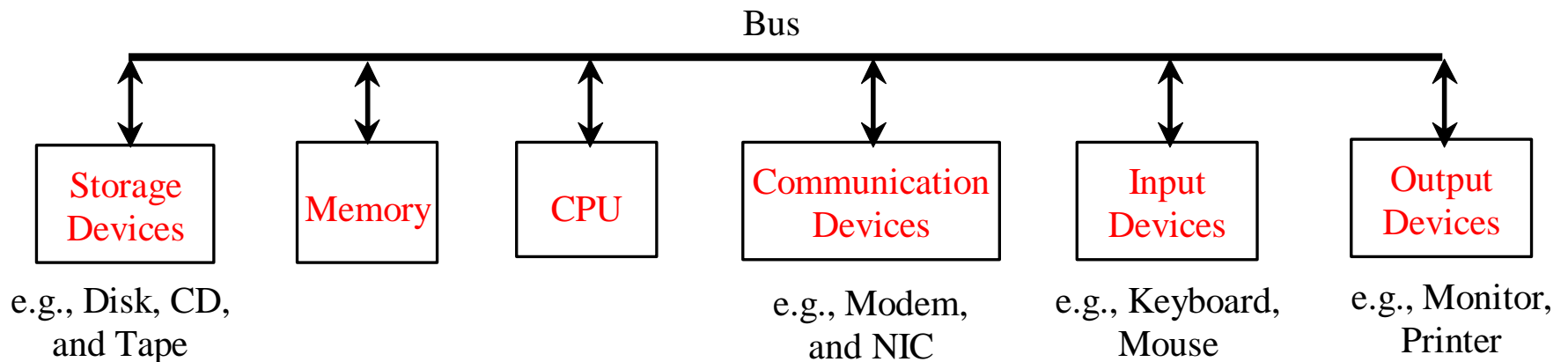
Introduction to Computers, Programs, and Python

Objectives

- ➡ To understand computer basics, programs, and operating systems
- ➡ To write and run a simple Python program
- ➡ To explain the basic syntax of a Python program
- ➡ To explain the importance of, and provide examples of, proper programming style and documentation
- ➡ To explain the differences between syntax errors, runtime errors, and logic errors
- ➡ To create a basic graphics program using Turtle

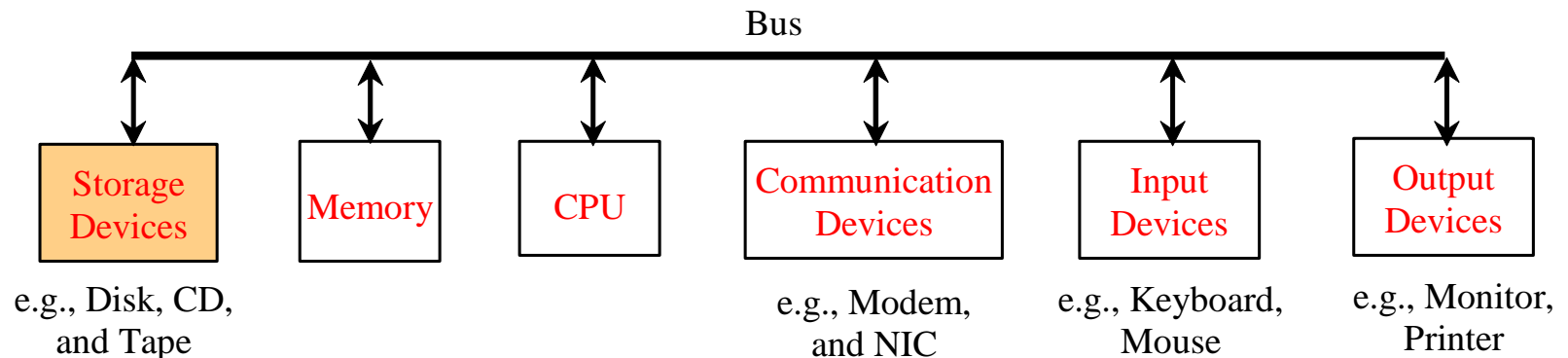
What is a Computer?

A computer consists of a CPU, memory, hard disk, floppy disk, monitor, printer, and communication devices.



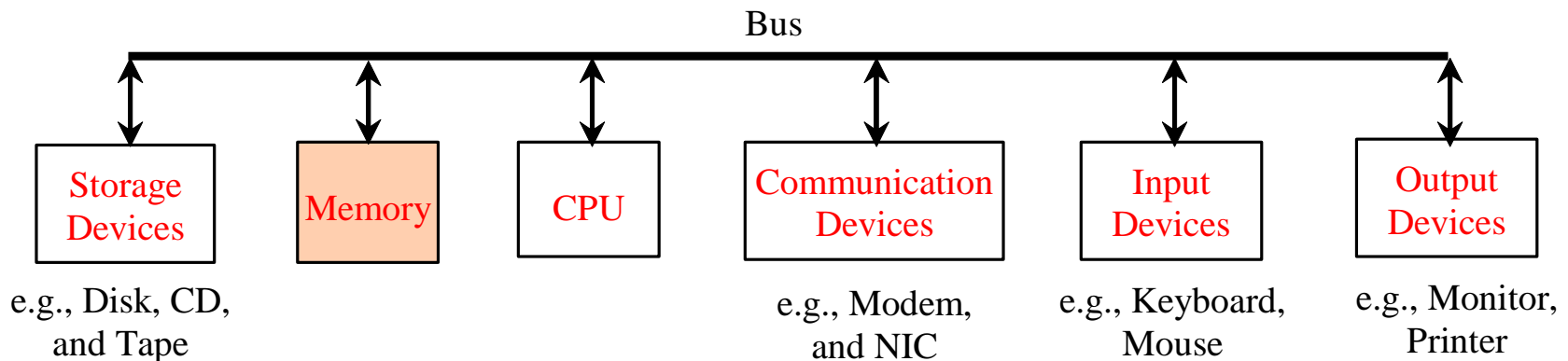
Storage Devices

Memory is volatile, because information is lost when the power is off. Programs and data are permanently stored on storage devices and are moved to memory when the computer actually uses them. There are three main types of storage devices: disk drives (hard disks and *floppy disks*), CD/DVD drives, and tape drives.



Memory

Memory is to store data and program instructions for CPU to execute. A memory unit is an ordered sequence of bytes, each holds eight bits. A program and its data must be brought to memory before they can be executed. A memory byte is never empty, but its initial content may be meaningless to your program. The current content of a memory byte is lost whenever new information is placed in it.



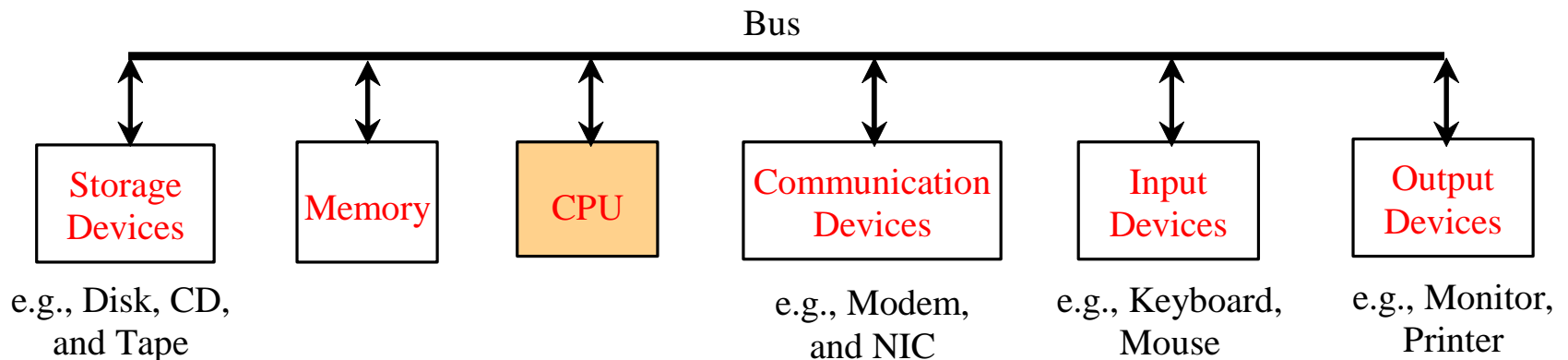
How Data is Stored?

- ▶ Data of various kinds, such as numbers, characters, and strings, are encoded as a series of bits (zeros and ones).
- ▶ Computers use zeros and ones because digital devices have two stable states; off (zero) and on (one).
- ▶ No two data items can share or split a same byte. A byte is the minimum storage unit.

Memory address	Memory content	
.	.	
.	.	
.	.	
2000	01001010	Encoding for character 'J'
2001	01100001	Encoding for character 'a'
2002	01110110	Encoding for character 'v'
2003	01100001	Encoding for character 'a'
2004	00000011	Encoding for number 3

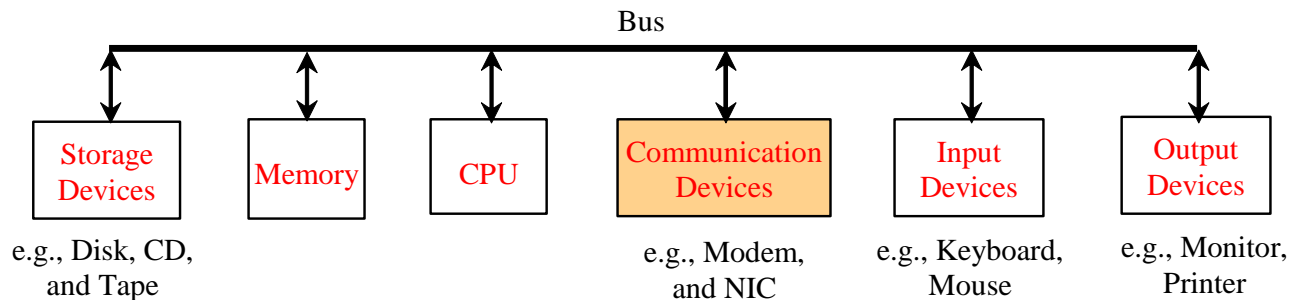
CPU

The central processing unit (CPU) is the brain of a computer. It retrieves instructions from memory and executes them. The CPU speed is measured in gigahertz (GHz), with 1 gigahertz equaling 1 billion pulses per second. The speed of the CPU has been improved continuously. If you buy a PC now, you can get an Intel i7 Processor at 4+ gigahertz.



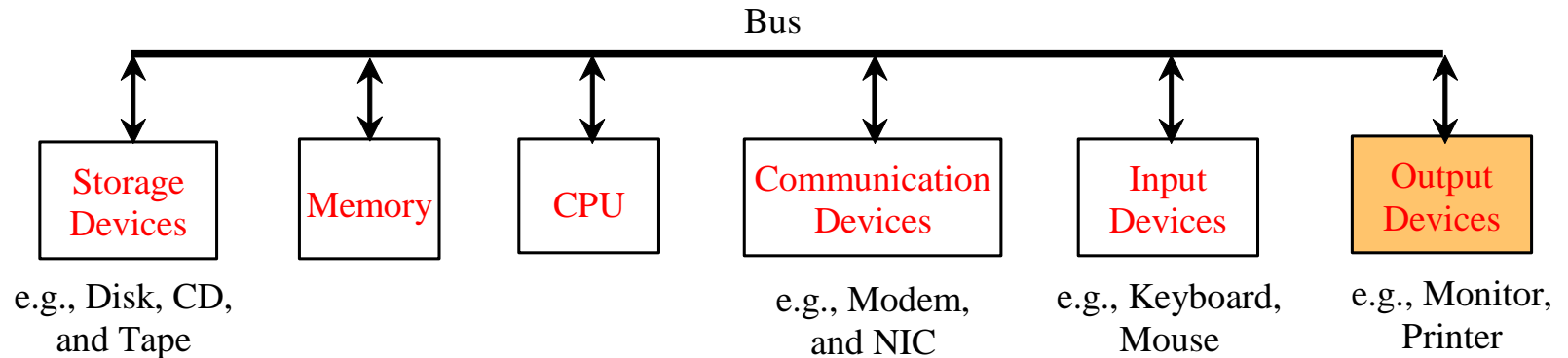
Communication Devices

A *regular modem* uses a phone line and can transfer data in a speed up to 56,000 bps (bits per second). A *DSL* (digital subscriber line) also uses a phone line and can transfer data in a speed 20 times faster than a regular modem. A *cable modem* uses the TV cable line maintained by the cable company. A cable modem is as fast as a DSL. Network interface card (*NIC*) is a device to connect a computer to a local area network (LAN). The LAN is commonly used in business, universities, and government organizations. A typical type of NIC, called *10BaseT*, can transfer data at 10 mbps (million bits per second).



Input Devices: Keyboard/Mouse

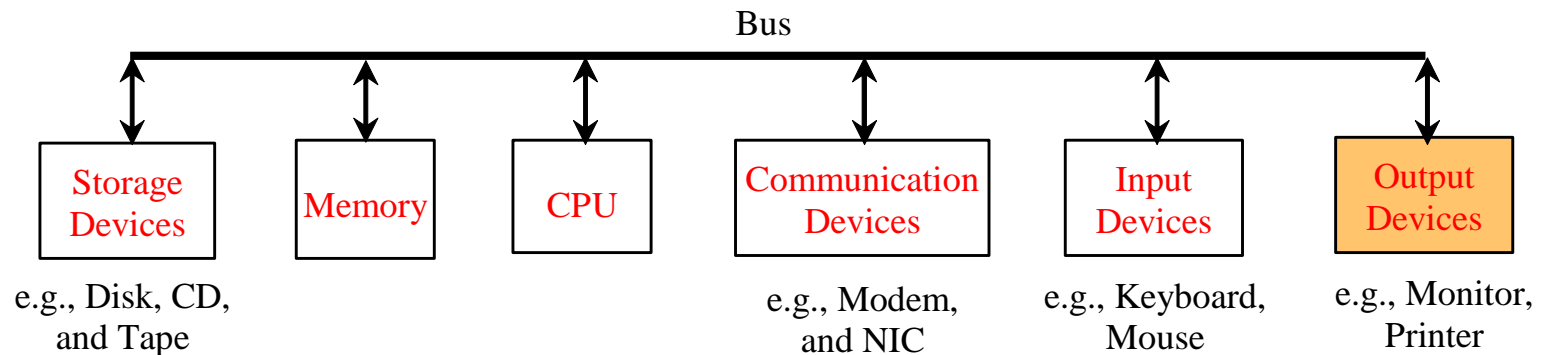
The keyboard and mouse allow the user to enter data and instructions into the computer.



Input Device: Touchpad
Input Device: Microphone

Output Devices: Monitor

The monitor displays information (text and graphics). The resolution and dot pitch determine the quality of the display.



Programs

Computer *programs*, known as *software*, are instructions to the computer.

You tell a computer what to do through programs. Without programs, a computer is an empty machine. Computers do not understand human languages, so you need to use computer languages to communicate with them.

Programs are written using programming languages.

Programming Languages

Machine Language

Assembly Language

High-Level Language

Machine language is a set of primitive instructions built into every computer. The instructions are in the form of binary code, so you have to enter binary codes for various instructions. Program with native machine language is a tedious process. Moreover the programs are highly difficult to read and modify. For example, to add two numbers, you might write an instruction in binary like this:

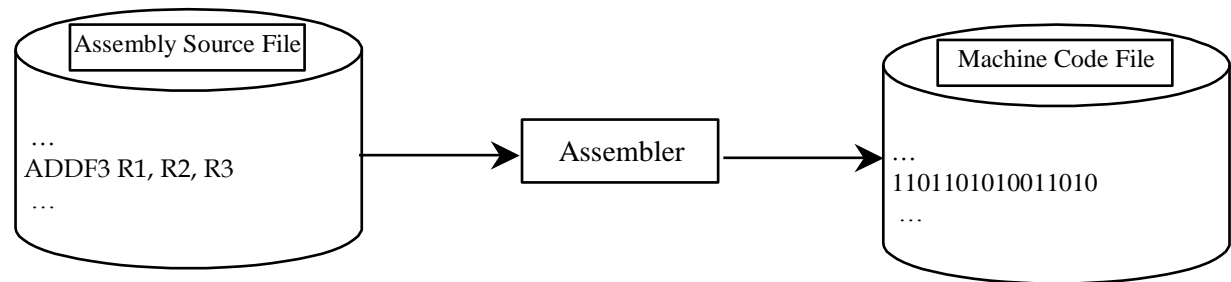
```
1101101010011010
```

Programming Languages

Machine Language Assembly Language High-Level Language

Assembly languages were developed to make programming easy. Since the computer cannot understand assembly language, however, a program called an assembler is used to convert assembly language programs into machine code. For example, to add two numbers, you might write an instruction in assembly code like this:

```
ADDF3 R1, R2, R3
```



Programming Languages

Machine Language Assembly Language High-Level Language

The high-level languages are English-like and easy to learn and program. For example, the following is a high-level language statement that computes the area of a circle with radius 5:

```
area = 5 * 5 * 3.1415;
```

Popular High-Level Languages

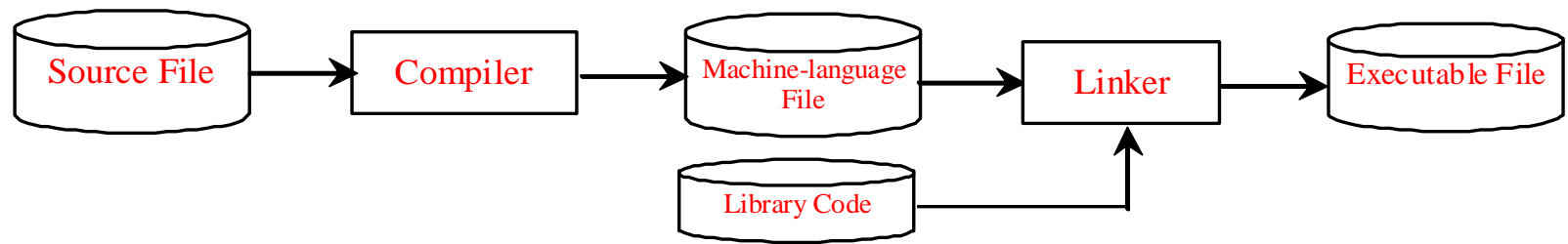
- ▶ COBOL (COmmon Business Oriented Language)
- ▶ FORTRAN (FORmula TRANslation)
- ▶ Pascal (named for Blaise Pascal)
- ▶ Ada (named for Ada Lovelace)
- ▶ C (whose developer designed B first)
- ▶ C++ (an object-oriented language, based on C)
- ▶ C# (a Python-like language developed by Microsoft)
- ▶ Java (language used in CMPS 260)
- ▶ Python (We use it in this class/book)

Compiling Source Code

A program written in a high-level language is called a *source program*.

Since a computer cannot understand a source program, a program called a *compiler* or *interpreter* is used to translate the source program into a machine language program called an *object* or *executable program*.

The object program is often then linked with other supporting library code before the object can be executed on the machine.



What is Python?

General Purpose

Interpreted

Object-Oriented

Python is a general purpose programming language. That means you can use Python to write code for any programming tasks.

Python are now used in:

- Google search engine
- mission critical projects in NASA
- processing financial transactions at New York Stock Exchange

What is Python?

General Purpose

Interpreted

Object-Oriented

Python is interpreted.

This means that Python code is translated and executed by an interpreter one statement at a time. In a compiled language, the entire source code is compiled and then executed altogether.

What is Python?

General Purpose

Interpreted

Object-Oriented

Python is an object-oriented programming language.

Object-oriented programming is a powerful tool for developing reusable software.

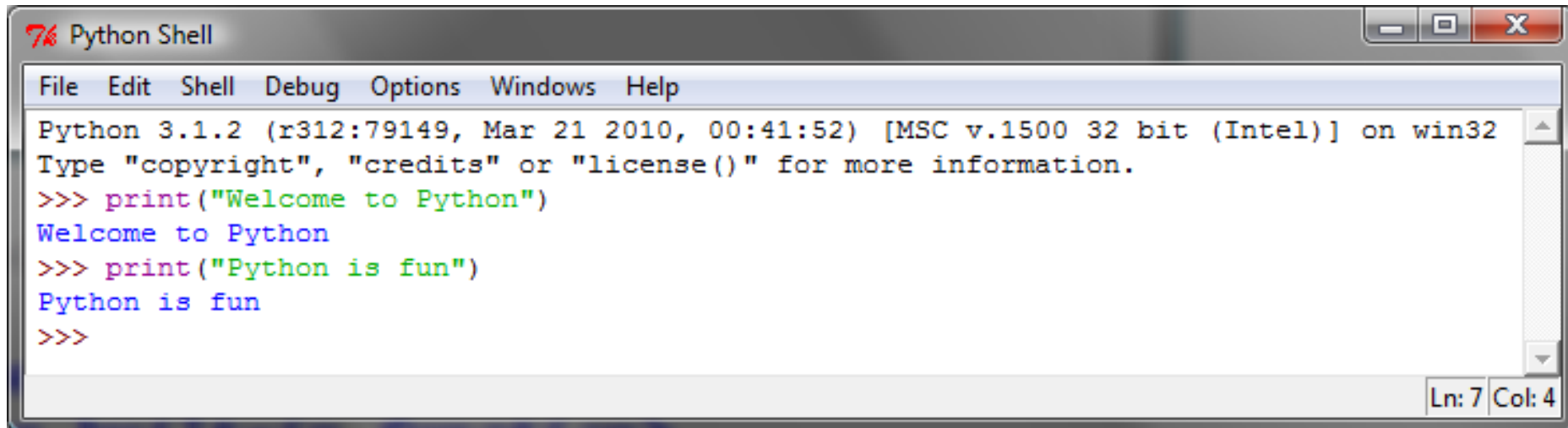
Python 2 vs. Python 3

Python 3 is a newer version, but it is not backward compatible with Python 2.

That means if you write a program using Python 2, it may not work on Python 3.

We will be using Python 3.

Launch Python IDLE



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.1.2 (r312:79149, Mar 21 2010, 00:41:52) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Welcome to Python")
Welcome to Python
>>> print("Python is fun")
Python is fun
>>>
Ln: 7 Col: 4
```

A Simple Python Program

Listing 1.1

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

Trace a Program Execution

Execute a statement

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

Trace a Program Execution

Execute a statement

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


Two More Simple Examples

WelcomeWithThreeMessages

ComputeExpression

Supplements on the Companion Website

www.cs.armstrong.edu/liang/py

Anatomy of a Python Program

- ▶ Statements
- ▶ Comments
- ▶ Indentation

Statement

A statement represents an action or a sequence of actions. The statement

`print("Welcome to Python")` in the program in

Listing 1.1 is a statement to display the greeting "Welcome to Python".

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

Indentation

The indentation matters in Python. Note that the statements are entered from the first column in the new line. It would cause an error if the program is typed as follows:

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

Special Symbols

Character Name		Description
()	Opening and closing parentheses	Used with functions.
#	Pound sign	Precedes a comment line.
" "	Opening and closing quotation marks	Enclosing a string (i.e., sequence of characters).
''' '''	Opening and closing quotation marks	Enclosing a paragraph comment.

Programming Style and Documentation

- ▶ Appropriate Comments
- ▶ Proper Indentation and Spacing Lines

Appropriate Comments

Include your name, class section, instructor, date, and a brief description at the beginning of the program.

Also include a summary at the beginning of the program to explain what the program does, its key features, its supporting data structures, and any unique techniques it uses.

Proper Indentation and Spacing

▶ Indentation

- When indentation is needed, indent 3 spaces.
- A consistent spacing style makes programs clear and easy to read, debug, and maintain.

▶ Spacing

- Use blank line to separate segments of the code.

Programming Errors

- ▶ Syntax Errors
 - Error in code construction
- ▶ Runtime Errors
 - Causes the program to abort
- ▶ Logic Errors
 - Produces incorrect result

Turtle Graphics

A simple way to start graphics programming is to use the Python built-in Turtle package.

OlympicSymbol