Programming Fundamentals (SWE – 102)

Introduction

Things you need to know …

- Instructor:
  - Engr. Sarwat Fawwad
- Office : BS:02
- Email: sfawwad@ssuet.edu.pk

Things you need to know …

- Textbook:
  - Introduction to Programming using Python
    - Y. Daniel Liang
- Reference Books:
  - Python Crash Course
    - Eric Matthes
  - Head First Python
    - Paul Barry
- Online Sources:
  - https://realpython.com/
  - many many more …

Marks Distribution

Theory

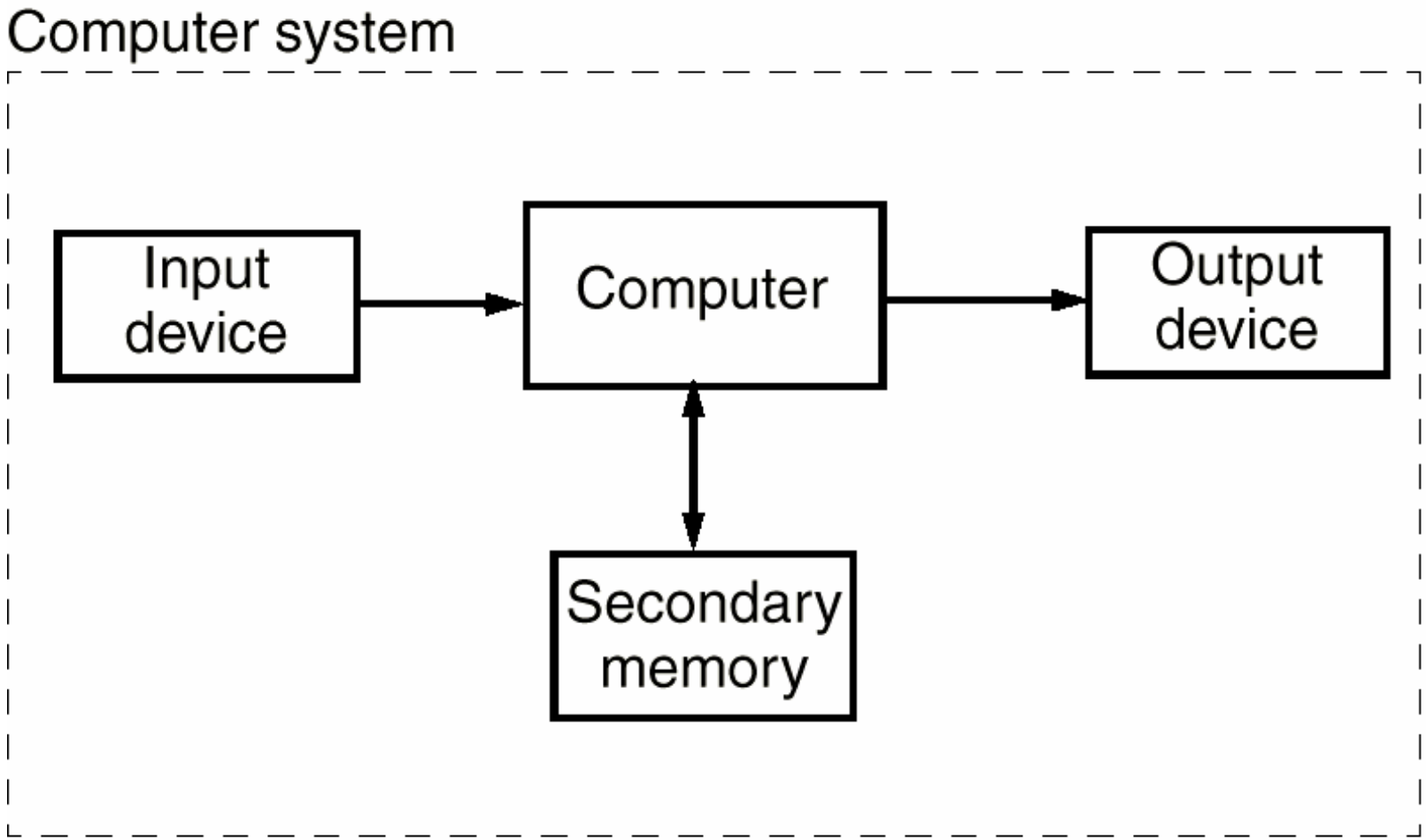
Quizes	10 Marks
Assignments	10 Marks
Midterm Examination	30 Marks
Final Examination	50 Marks
Total	**100 Marks**

Laboratory

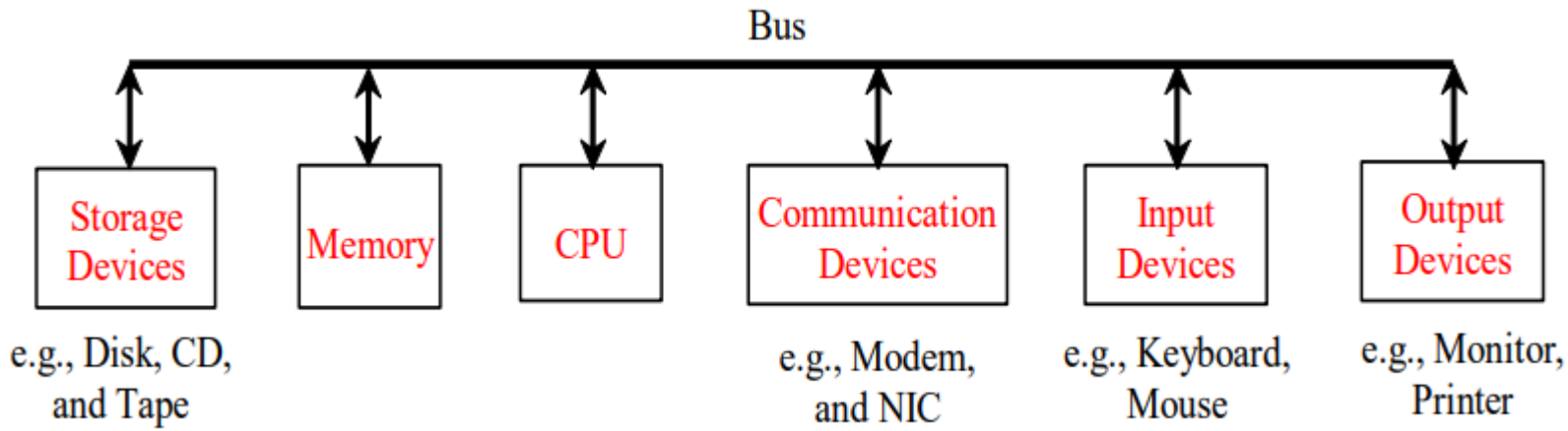
Quizes /Assignments / Lab File	15 Marks
Project	15 Marks
Final Exam	20 Marks
Total	**50 Marks**

What is a Computer ?

- A Computer is a smart or intelligent machine, which can work only on certain instructions given by a human being.
- A data processing machine operated automatically by using of a list of the instructions (called a program) stored in its main memory.
- Collects data (input)
- Processing
- Produces information (output)

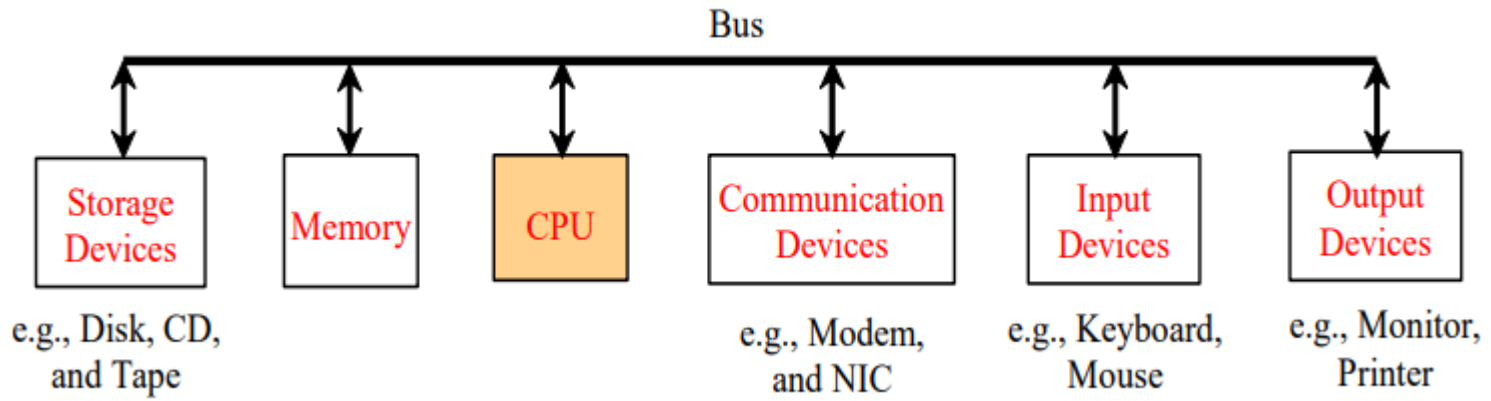

Computer system
Input device
Computer
Output device
Secondary memory

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



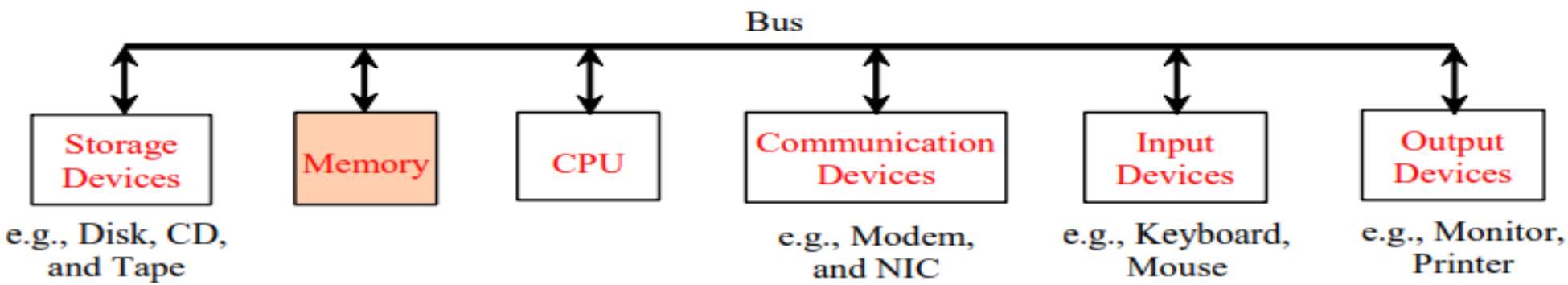
CPU

- The central processing unit (CPU) is the brain of a computer. It retrieves instructions from memory and executes them.



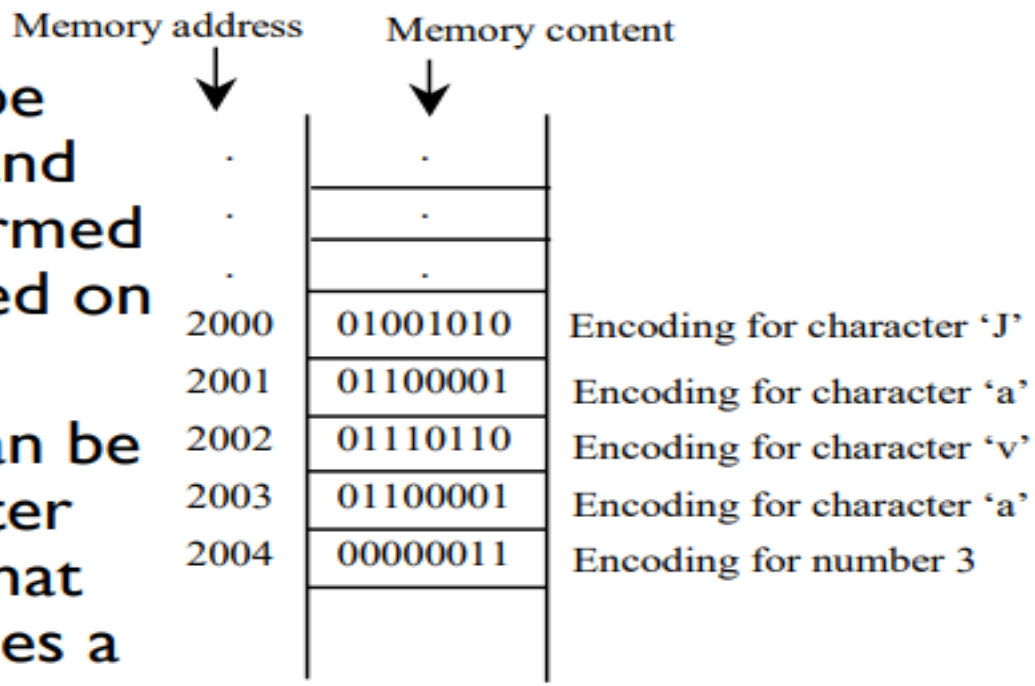
Memory

- Memory is to store data and program instructions for CPU to execute.
- A memory unit is an ordered sequence of bytes, each holds 8 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.



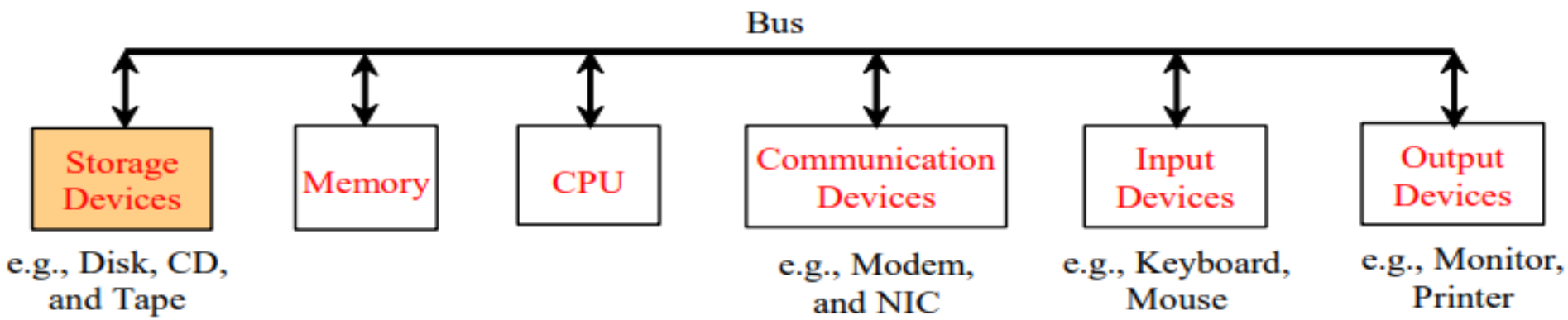
How Data is Stored?

- Data of various kinds, are encoded as a series of bits (zeros and ones).
- The programmers need not to be concerned about the encoding and decoding of data, which is performed automatically by the system based on the encoding scheme.
- A small number such as three can be stored in a single byte. If computer needs to store a large number that cannot fit into a single byte, it uses a number of adjacent bytes.
- No two data can share or split a same byte. A byte is the minimum storage unit.



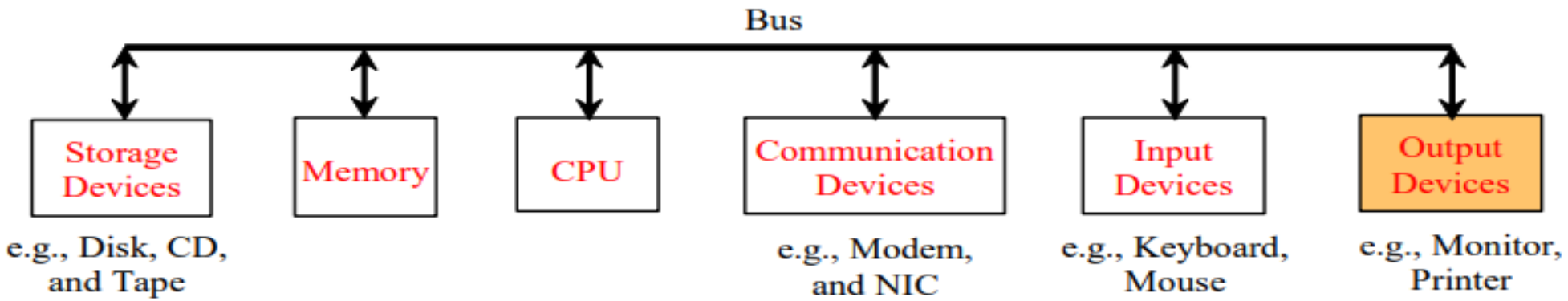
Storage Devices

- 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.



Output Devices: Monitor

- The monitor displays information (text and graphics).
- **The resolution and dot pitch determine the quality of the display.**
- The higher a monitor's resolution is, the more detailed an image can be
- because a higher resolution monitor will be made up of more **pixels** than a lower resolution monitor

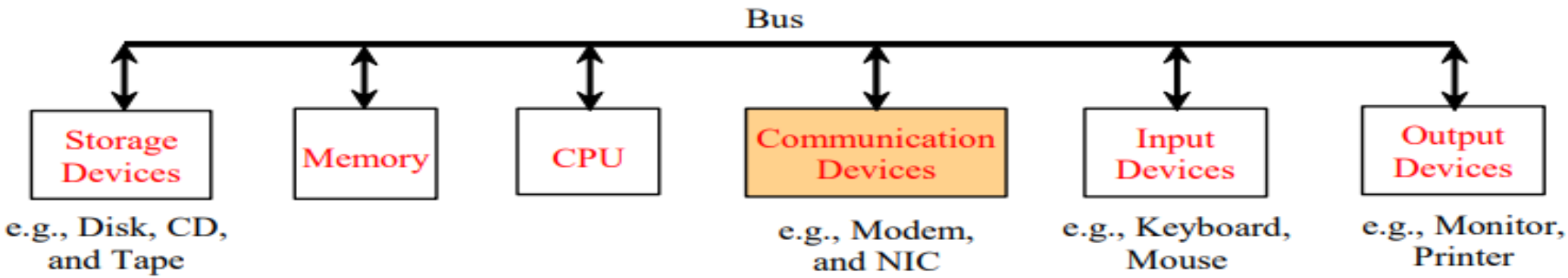


Monitor Resolution and Dot Pitch

- **Resolution**
  - The *resolution* specifies the number of pixels per square inch.
  - Pixels (short for "picture elements") are tiny dots that form an image on the screen.
  - The higher the resolution, the sharper and clearer the image is.
  - PC monitors are usually 15-inch, 17-inch, 19-inch, or 21-inch.

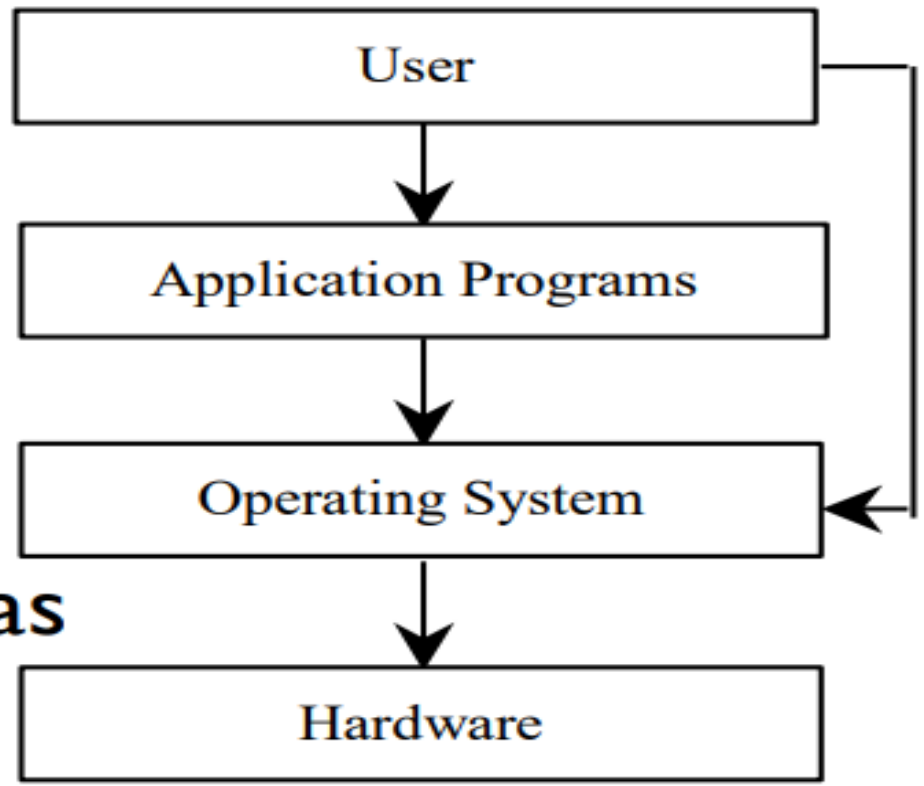
Communication Devices

- Network interface card (*NIC*) is a device to connect a computer to a local area network.
- The LAN is commonly used in business, universities, and government organizations.
- A *DSL* (digital subscriber line) uses a phone line to transfer data.



Operating Systems

- The operating system (OS) is a program that manages and controls a computer's activities e.g. Windows 10.
- Windows is currently the most popular PC operating system.
- Application programs such as an Internet browser and a word processor cannot run without an operating system.



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:

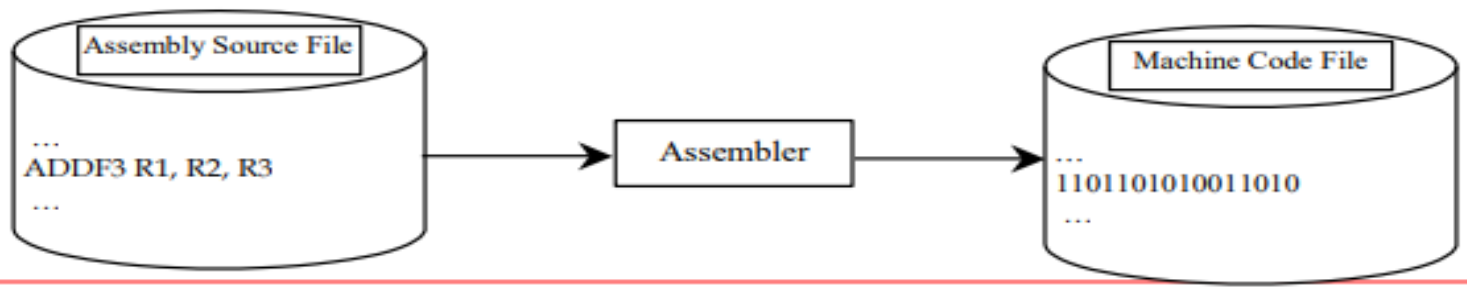
```
1101101010011010
```

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 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:

```
ADD 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;
```

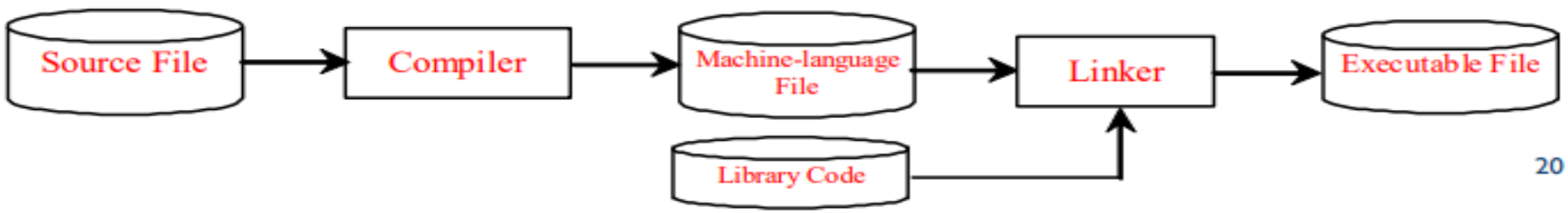
```
area = π r ^2
```

Popular High-Level Languages

- COBOL (COmmon Business Oriented Language)
- FORTRAN (FORmula TRANslation)
- BASIC (Beginner All-purpose Symbolic Instructional Code)
- Pascal (named for Blaise Pascal)
- C (whose developer designed B first)
- Visual Basic (Basic-like visual language developed by Microsoft)
- Delphi (Pascal-like visual language developed by Borland)
- C++ (an object-oriented language, based on C)
- C# (a Python-like language developed by Microsoft)
- Python (We use it in this course)

Compiling Source Code

- A program written in a high-level language is called a source program.
- Since a computer cannot understand a source program.
- Program called a compiler is used to translate the source program into a machine language program called an object program.
- The object program is often then linked with other supporting library code before the object can be executed on the machine.



Generations of Programming Language

- The first generation languages, or 1GL, are low-level languages that are machine language.
- The second generation languages, or 2GL, are also low-level languages that generally consist of assembly languages.
- The third generation languages, or 3GL, are high-level languages such as C.

Generations of Programming Language

- The fourth generation languages, or 4GL, are languages that consist of statements similar to statements in a human language. Fourth generation languages are commonly used in database programming and scripts.
- The fifth generation languages, or 5GL, are programming languages that contain visual tools to help develop a program. A good example of a fifth generation language is Visual Basic.

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 is now used in Google search engine, in mission critical projects in NASA, in processing financial transactions at New York Stock Exchange.

What is Python?

General Purpose Interpreted Object-Oriented

Python is interpreted, which 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.

Data in Python are objects created from classes.

A class is essentially a type that defines the objects of the same kind with properties and methods for manipulating objects.

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

Python is Dynamic Language

- **Python is a Dynamic Language:** Python is a dynamic language for the beginner to advance level programmers and supports the development of a wide range of applications include
- Simple text processing applications
- Functional applications
- Object-Oriented base applications
- Web applications (flask and django)
- GUI base applications
- Artificial Intelligence base applications
- Interactive Gaming programming.

Python's History

- Created by Guido van Rossum in Netherlands in 1990
- Open source
- 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.

Popular Python IDEs

For Beginners

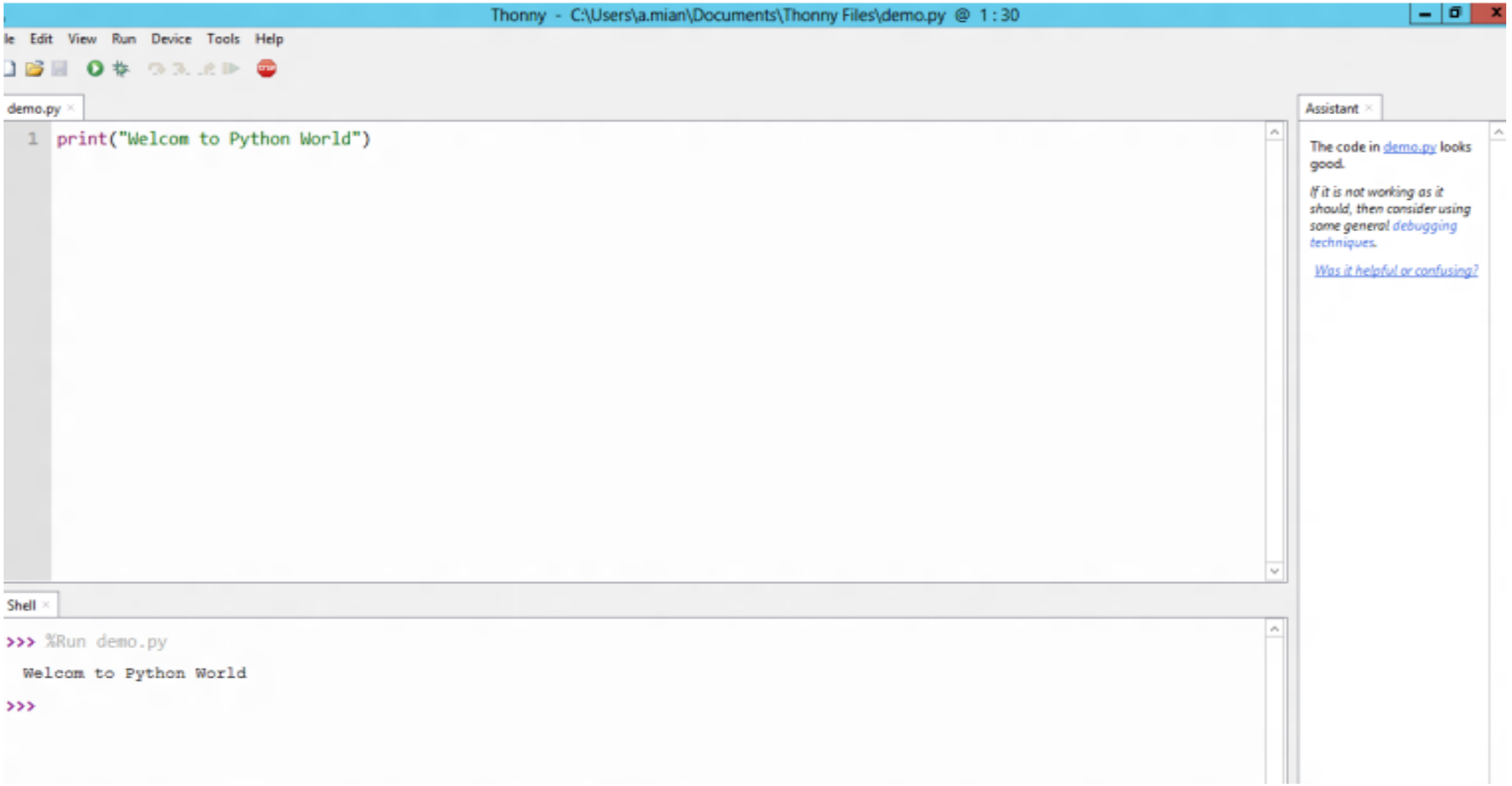
- IDLE
- Thonny

For Professional Developers

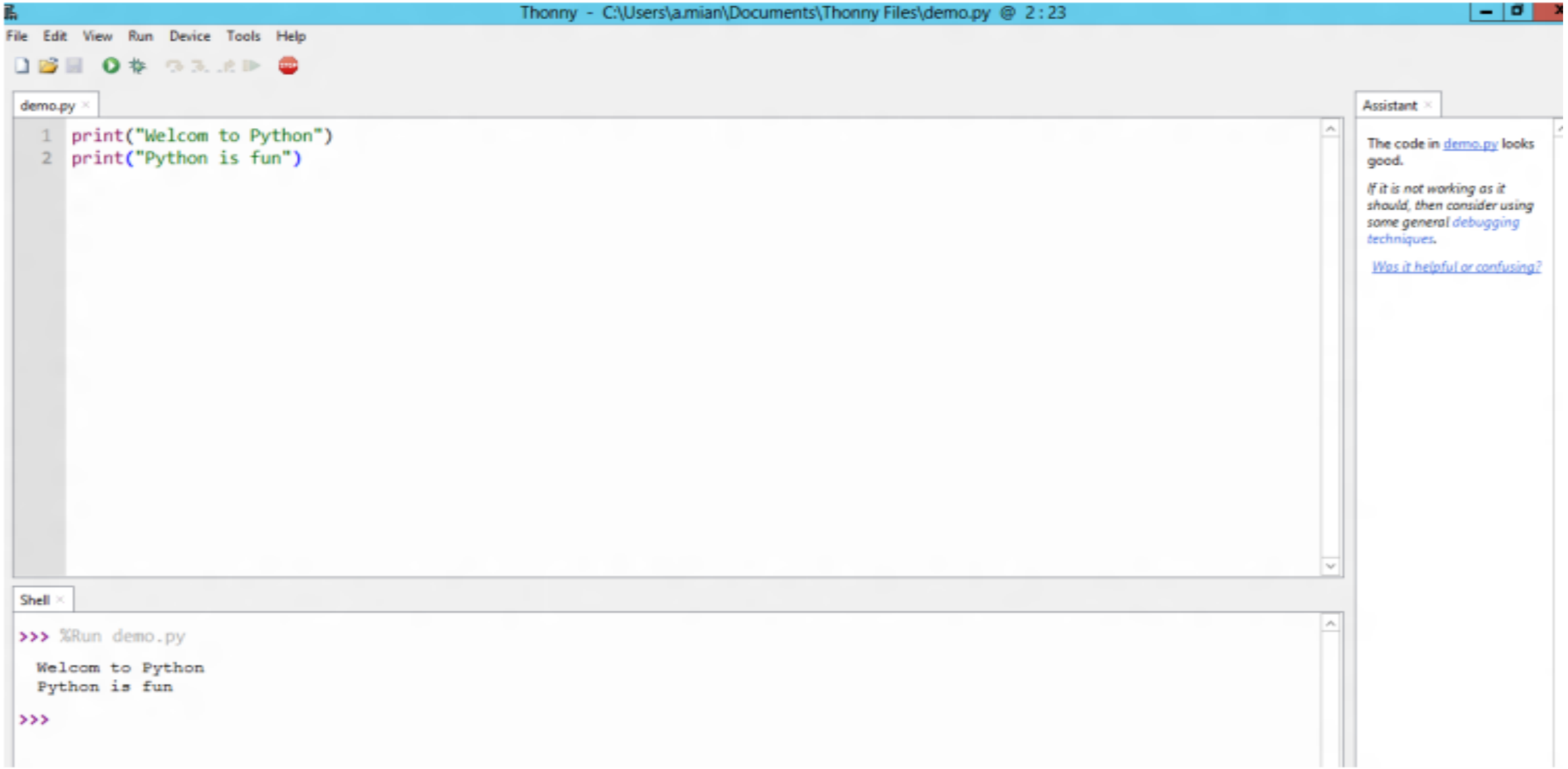
- Spyder
- Jupyter
- PyCharm
- Visual Studio

We will use Thonny in our course…

Thonny Environment



Run Python Program



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") 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")
```

Appropriate Comments

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

Proper Indentation and Spacing

- Indentation
  - Indent four 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.

```
print(3+4*4) bad style
print(3 + 4 * 4) good style
```

Programming Errors

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

  (5 / 9 * 35 – 32) = -12.555

  5 / 9 * (35 - 32) =1.66

Summary

- Computer
- Programming Languages
- Python Language
- Programming with Python