



COMP 1023 Introduction to Python Programming Introduction to Computers and Programming

Dr. Cecia Chan, Prof. SC Cheung, Dr. Alex Lam, Dr. Desmond Tsoi

Department of Computer Science & Engineering
The Hong Kong University of Science and Technology, Hong Kong SAR, China



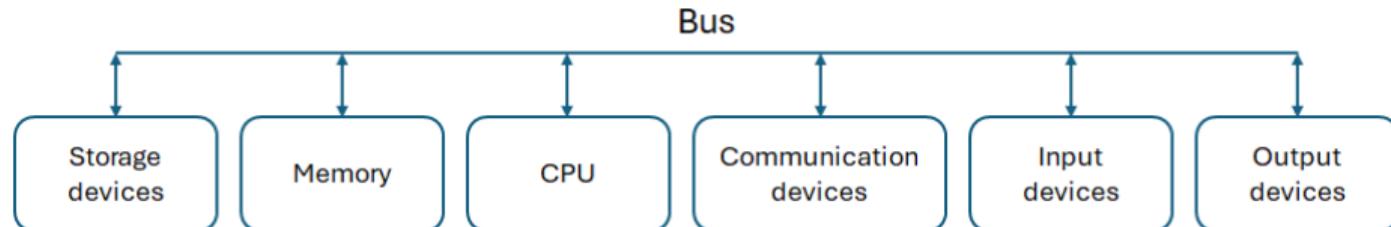
Part I

Understanding Computers and Basic Concepts



What is a Computer?

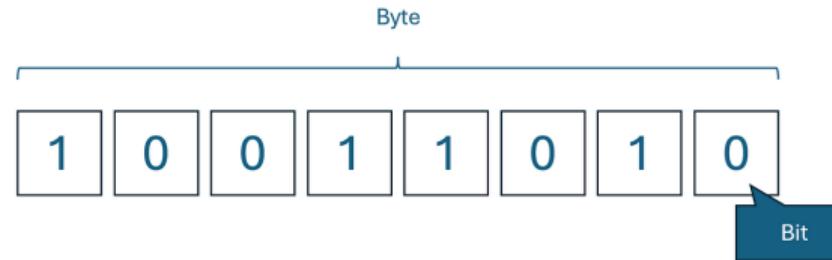
- A **computer** consists of both **hardware and software**.
- **Hardware** includes the visible, **physical components** of the computer:
 - Central Processing Unit (CPU)
 - Main Memory (RAM)
 - Storage Devices (e.g., hard drives, SSDs)
 - Input Devices (e.g., keyboard, mouse)
 - Output Devices (e.g., monitors, printers)
 - Communication Devices (e.g., network interface cards)
- All components of a computer are interconnected by a bus.



- **Software** provides the **invisible instructions** that control the hardware and enable it to perform specific tasks.

Bits and Bytes

- Information (**data and programs**) is stored in a computer as **sequences of 0s and 1s**.
- These 0s and 1s are interpreted as digits in the binary number system and are called **bits** (binary digits).
- A **byte** is composed of **8 bits**.



- A computer's storage capacity is measured in bytes and multiples of bytes:
 - A kilobyte (KB) is 1,024 bytes.
 - A megabyte (MB) is 1,024 KB, which is $1,024 \times 1,024$ bytes.
 - A gigabyte (GB) is 1,024 MB, which is $1,024 \times 1,024 \times 1,024$ bytes.
 - A terabyte (TB) is 1,024 GB, which is $1,024 \times 1,024 \times 1,024 \times 1,024$ bytes.

Memory

- A computer's **memory** consists of an **ordered sequence of bytes** used for storing programs and the data that the programs are working with.
- **Each byte** in memory has a **unique address**, which is used to locate the byte for storing and retrieving data.

Memory address	Memory content	
.	.	
.	.	
.	.	
2000	01000011	Encoding for character C
2001	01110010	Encoding for character r
2002	01100101	Encoding for character e
2023	01110111	Encoding for character w
2024	00000011	Decimal number 3



Part II

Problem Solving, Algorithms & Computer Programming Languages



Problem:

e.g. I want to write an image processing software, how to do it? :P

Algorithm:

A step-by-step description on how to solve the problem

Implementation:

Write programs according to the algorithm description using a programming language

Problem Solving

Problem Solving

- There are many real-world problems that need to be solved to make our lives easier.
- These problems are typically solved using computers, as they can perform operations much faster than humans can.
- However, computers are intelligent only if humans can **instruct/teach them to perform relevant tasks**.
- To do this, a **step-by-step description** of how to accomplish a task is needed.
 - We call this description an “**algorithm**”.

Some Real World Multimedia Problems:



Algorithm and its Representations

- A step-by-step description can be represented in many different ways:
 - Flowcharts
 - Pseudocode
 - Computer programs



Example

- Problem: Calculate the discount rate for customers in a retail store.

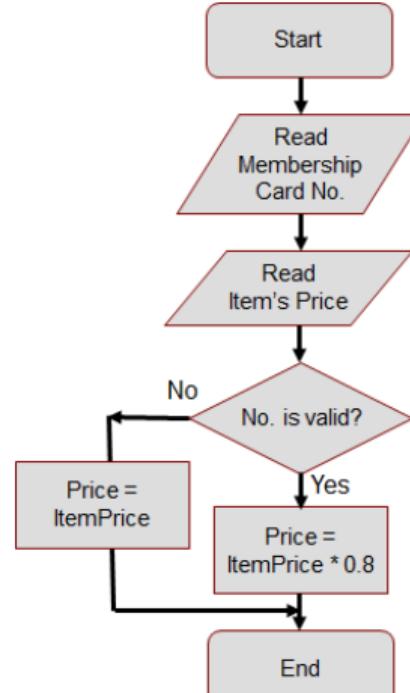


Algorithm Representation: Flowchart and Pseudocode



```
READ Membership Card No.  
READ ItemPrice  
IF Membership Card No. is valid THEN  
    price = ItemPrice * 0.8;  
ELSE  
    price = ItemPrice;  
END IF
```

Pseudocode



Flowchart

Algorithm Representation: Computer Program(s)

```
# Filename: sample_python_program.py
def is_valid(num):
    valid = False
    if num > 0 and num <= 100:
        valid = True
    return valid

card_no = int(input("Enter the card number: "))
item_price = int(input("Enter the item price: "))
price = 0
if is_valid(card_no):
    price = item_price * 0.8
else:
    price = item_price
print("The price is", price)
```

Computer program written in Python

Not a good representation of an algorithm, as different programmers may use different programming languages to solve the problem.

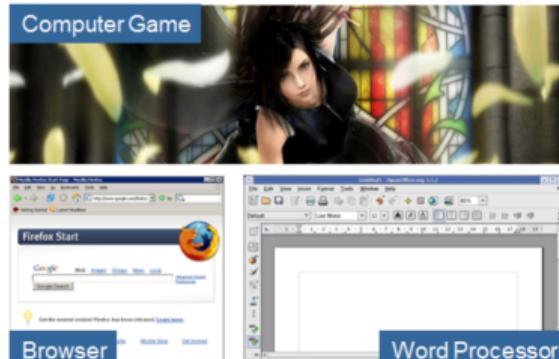


Computer Programs and Programming

- Once we come up with an algorithm, the next step is to write a computer program according to the steps in the algorithm.
- A computer program:
 - Set of step-by-step instructions
 - Performs specific tasks to solve a problem
 - Examples:
 - Computer games
 - Word processors
 - Web browsers
 - ...
- Computer programming is the design and implementation of computer programs.

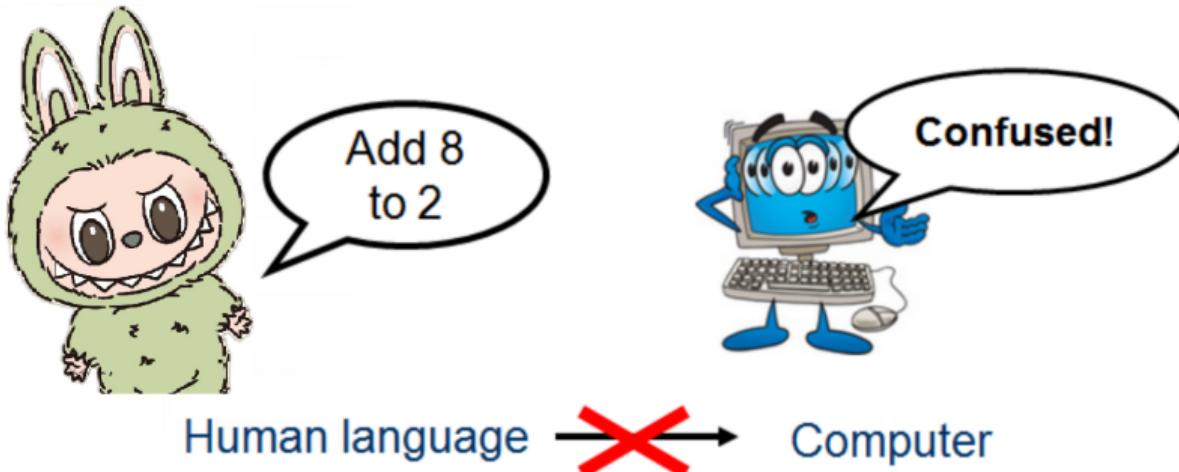
```
def is_valid(num):
    valid = False
    if num > 0 and num <= 100:
        valid = True
    return valid

card_no = int(input("Enter the card number: "))
item_price = int(input("Enter the item price: "))
price = 0
if is_valid(card_no):
    price = item_price * 0.8
else:
    price = item_price
print("The price is", price)
```



Computer Programming

- Learning computer programming is similar to learning a natural language such as English, Japanese, Korean, etc.
- Although they are similar, it doesn't mean that they are exactly the same.



- Don't worry! This is actually good. There is a more systematic way to learn computer programming, making it much easier to learn, in my opinion. ;) (Good news!)

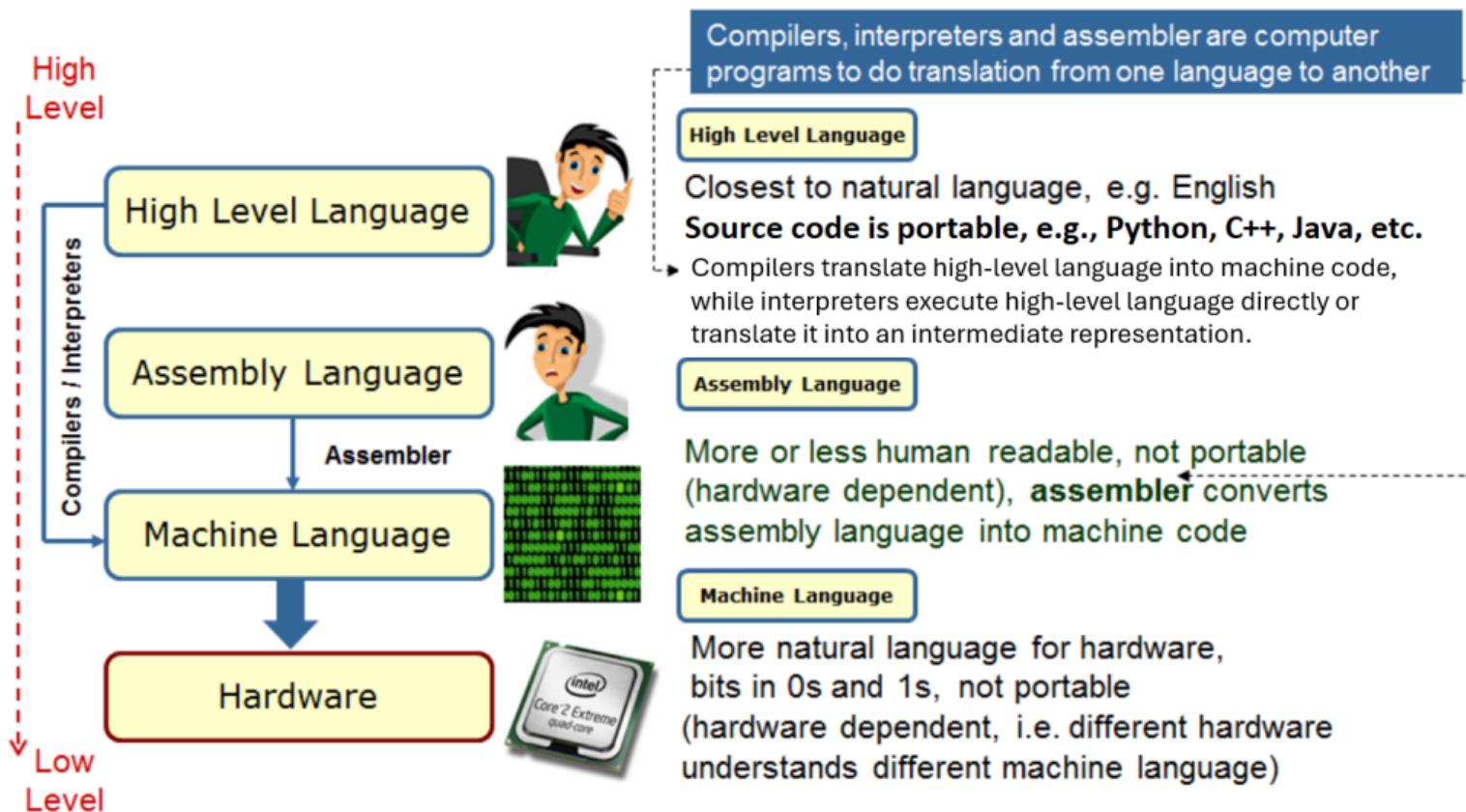
Programming Languages

- Computer programs are written in programming languages.
- Unlike human languages, a **programming language** defines **a set of instructions in a specific format** that can be given to a computer.
- Two important issues when writing programs:
 1. **Program syntax** – Is the grammar of the instructions correct?
 2. **Program logic** – Is the program able to solve the problem?

We will talk more about these in the next set of notes! :)



Categorization of Programming Languages

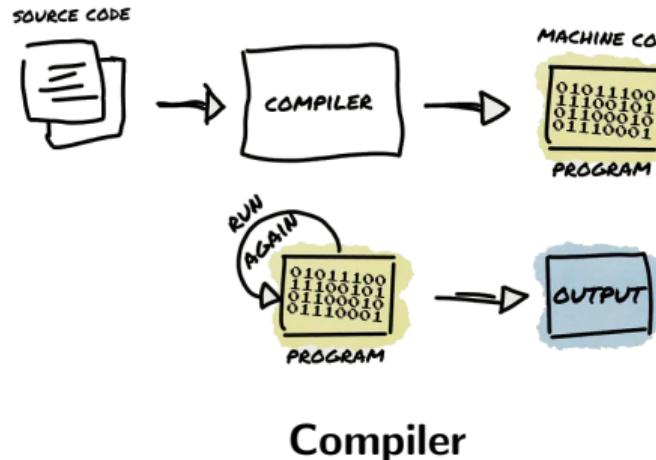


Compiler vs. Interpreter

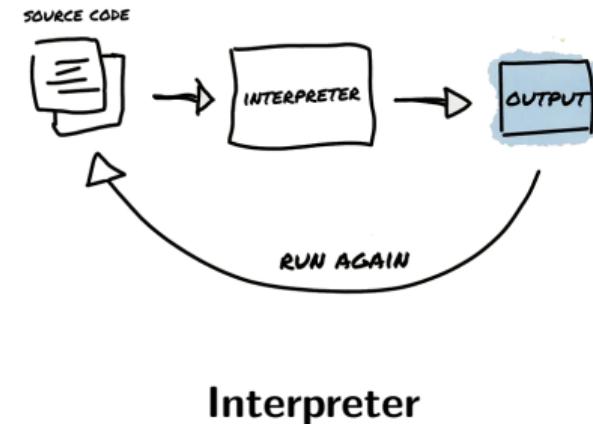
- A program written in a high-level language is called a **source program** or **source code**.
- A **compiler** translates the entire source code into a machine code file, which is then **executed**.
- An **interpreter** reads one statement from the source code, **executes it immediately**, and may **translate it** to an intermediate representation or machine code if necessary, before continuing to the next statement.



Compiler vs. Interpreter



Compiler



Interpreter

Pictures by mrs-th.github.io.

- **Compiler:** Translates the entire source code to machine code before execution.
- **Interpreter:** Translates and executes code line by line.

Speed: Compiled programs run faster than interpreted ones.

High-Level Languages

- High-level languages are programming languages that are user-friendly and abstract away most of the hardware details.
- They can mainly be divided into two categories:
 - Procedural or Structured Programming Languages
 - Examples: COBOL, Pascal, C, etc.
 - Object-Oriented Programming (OOP) Languages
 - Examples: Python, C++, Java, etc.



COBOL:
Grace Murray Hopper



Pascal:
Niklaus Wirth



C Language:
Dennis Ritchie



Python:
Guido van Rossum



Java:
James Gosling



C++:
Bjarne Stroustrup

Current Popularity of Programming Languages

Jun 2025	Jun 2024	Change	Programming Language	Ratings	Change
1	1		 Python	25.87%	+10.48%
2	2		 C++	10.68%	+0.65%
3	3		 C	9.47%	+0.24%
4	4		 Java	8.84%	+0.44%
5	5		 C#	4.69%	-1.96%
6	6		 JavaScript	3.21%	-0.11%
7	7		 Go	2.28%	+0.35%
8	9		 Visual Basic	2.20%	+0.54%
9	11		 Delphi/Object Pascal	2.15%	+0.62%
10	10		 Fortran	1.86%	+0.33%

TIOBE Index for June 2025: <https://www.tiobe.com/tiobe-index/>

Part III

Briefing on Python

“Remember to introduce me to
your students when you talk
about Python! :P”



Python: A Brief History

- Python is not only a type of large, non-venomous snake but also a popular programming language. :D
- History of the Python programming language:
 - Origins (late 1980s): Created by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands.
 - Initial Release (1991): Python 1.0 was first released to the public in 1991.
 - Growing Popularity (1990s): Gained traction as a general-purpose programming language, especially in web development, scientific computing, and data analysis.
 - Major Versions: Significant releases included Python 2.0 (2000) and Python 3.0 (2008).
 - Widespread Adoption (2010s): Surged in popularity, particularly in data science, machine learning, and artificial intelligence.
 - Current State (2020s): One of the most widely used programming languages today.



Major Python Versions

- Python 2.x
 - No longer supported as of January 1, 2020. No new bug reports, fixes, or changes will be made.
- Python 3.x
 - Not backwards compatible with Python 2.x.

Latest Stable Version: [Python 3.13.](#)

Python 3.13 Documentation: <https://docs.python.org/3/>



Python

- **Easy to Learn:** Python has a simple syntax, making it beginner-friendly and easy to start coding.
- **Interpreted:** Python is an interpreted language, meaning code is executed line-by-line, allowing for **interactive testing** of small code snippets.
- **Versatile:** Python is **suitable for a wide range of applications**, including web development, data analysis, artificial intelligence, and more.
- **Large Standard Library:** Python includes a vast standard library that **provides extensive functionality** out-of-the-box.
- **Cross-Platform:** Python code runs on various operating systems, including Windows, macOS, and Linux.

Essential Tools for Python Program Development

To write Python programs, you need **two components**:

1. Machine with Python Software Development Kit (Python SDK) installed

(We use Python SDK version **3.13** for this course)

<https://www.python.org/downloads/>

The SDK includes:

- Python interpreter
- Python standard library
- IDLE (Integrated Development and Learning Environment)
- Python Package Installer (pip)
- Python Documentation
- Python Tools and Utilities

Essential Tools for Python Program Development

2. Python Integrated Development Environment (IDE):

Software with a code editor, interpreter integration, and additional features.

- Visual Studio Code (VS Code)
<https://code.visualstudio.com/>
download
- PyCharm
- Jupyter Notebook
- Atom
- Sublime Text
- Spyder

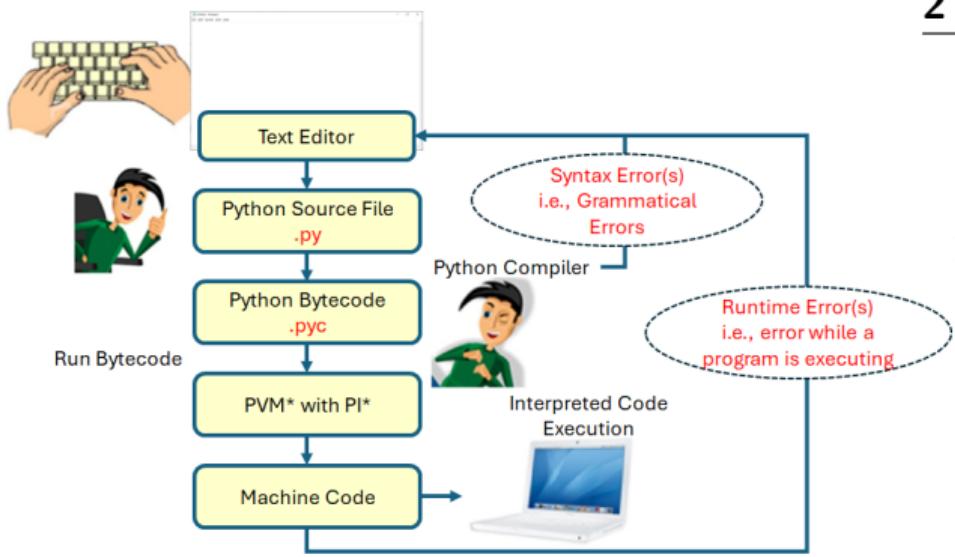
The screenshot shows the official Visual Studio Code download page. At the top, there's a navigation bar with links to Visual Studio Code, Docs, Updates, Blog, API, Extensions, FAQ, GitHub Copilot, and MCP. A search bar and a 'Download' button are also present. Below the navigation, there's a message: 'Try MCP servers to extend agent mode in VS Code!'. The main section is titled 'Download Visual Studio Code' with the subtext 'Free and built on open source. Integrated Git, debugging and extensions.' It features icons for Windows, Linux (Ubuntu), and macOS. Each platform has a row of download links:

- Windows: User Installer (.msi, .AppImage), System Installer (.exe, .AppImage), .zip, CLI (.msi, .AppImage)
- .deb: .deb, .rpm, .tar.gz, Snap, CLI (.deb, .rpm, .tar.gz, .AppImage)
- .rpm: .deb, .rpm, .tar.gz, Snap, CLI (.deb, .rpm, .tar.gz, .AppImage)
- Mac: .zip (Intel chip, Apple silicon, Universal), CLI (Intel chip, Apple silicon)

Visual Studio Code IDE

Please refer to the download section of our course website for the Python SDK and VS Code.

Development Cycle of a Python Program: Overview



2 Steps

1. **Write Python Source Code:** Use an editor to write your code and save it with a .py extension.
 2. **Run the Source File:** Execute the .py file. If there are any syntax or runtime errors, return to step 1!
Details:

Details:

- Bytecode compilation generates .pyc files.
 - Execute bytecode using the PVM.

* PVM stands for Python Virtual Machine, and PI stands for Python Interpreter.

Types of Programming Errors in Python

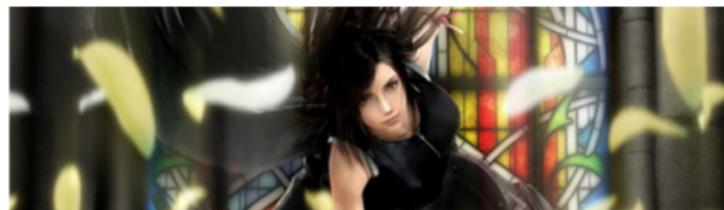
- **Syntax Errors:** Occur when the program **doesn't follow the syntax rules** of the language.
For example, forgetting a quotation mark or misspelling a keyword.
- **Runtime Errors:** Happen **during program execution** when Python encounters an **operation that can't be performed**, such as division by zero.
- **Logic Errors:** Occur when the **program runs successfully**, but it doesn't produce the desired outcome or **behaves unexpectedly**.



Part IV

First Python Program

Let's Start Writing Our First Python Program!



No! Start with a
simple one! :P

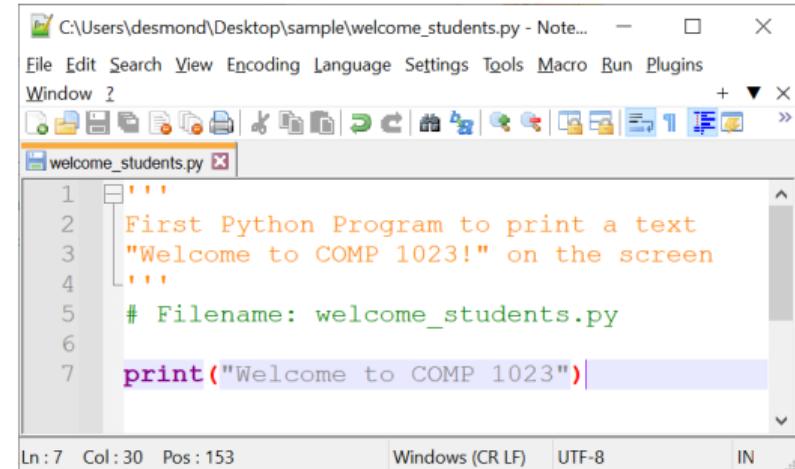


My First Python Program

Step 1: Write Python source code using an editor and save the file with a .py extension.

```
'''  
First Python Program to print a text  
"Welcome to COMP 1023!" on the screen  
# Filename: welcome_students.py  
  
print("Welcome to COMP 1023")
```

Note: Python is a **CASE-SENSITIVE LANGUAGE**, meaning it treats lower-case and upper-case letters differently!



A screenshot of a Windows Notepad window titled "welcome_students.py". The window contains the following Python code:

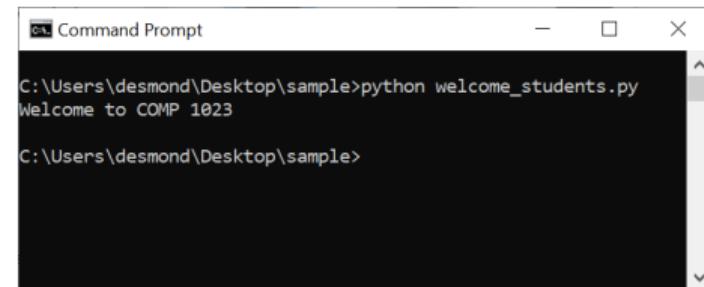
```
'''  
First Python Program to print a text  
"Welcome to COMP 1023!" on the screen  
# Filename: welcome_students.py  
  
print("Welcome to COMP 1023")
```

The code is color-coded: orange for strings, green for comments, and purple for the `print` keyword. The Notepad interface includes a menu bar with File, Edit, Search, View, Encoding, Language Settings, Tools, Macro, Run, Plugins, Window, and Help. Below the menu is a toolbar with various icons. The status bar at the bottom shows "Ln : 7 Col : 30 Pos : 153" and "Windows (CR LF) UTF-8 IN".

How to Run Your First Python Program

Step 2: To run your program "welcome_students.py" (For Windows)

- First, open a command prompt by holding the "Windows Key" and pressing "r".
- Change the directory to where your source file is located. For example, if your file is in:
C:\Documents and Settings\Desmond\Desktop\sample
Type:
cd C:\Documents and Settings\Desmond\Desktop\sample
- Then type: python welcome_students.py in the command prompt (where python refers to the Python SDK interpreter).



```
Command Prompt
C:\Users\desmond\Desktop\sample>python welcome_students.py
Welcome to COMP 1023
C:\Users\desmond\Desktop\sample>
```

Checking Python Version

If Python 2 is installed, running python in the command prompt (cmd) may use Python 2. Please check the Python version by executing `python --version`.

Common Question About Running Python Programs

Q: Am I supposed to run Python programs using commands?

- You can also run Python programs using [Integrated Development Environments \(IDEs\)](#) by simply clicking on the icons.
- However, it is encouraged to learn command-line execution, as not all machines have IDEs installed!



Key Terms

- Algorithms
- Assembler
- Assembly Language
- Bits and Bytes
- Bytecode
- Case-Sensitive Language
- Compiler
- Computer Programs
- Computer Programming
- High Level Language
- IDE (Integrated Development Environment)
- Interpreter
- Logic error
- Machine Language
- Memory and address
- Object-Oriented Programming
- PI (Python Interpreter)
- Procedural / Structured Programming
- Pseudo Code
- PVM (Python Virtual Machine)
- Runtime error
- SDK (Software Development Kit)
- Syntax error

Review Questions

Fill in the blanks in each of the following sentences about the Python environment.

1. A _____ is a binary digit 0 or 1.
2. A _____ is a sequence of 8 bits.
3. A _____ translates the entire source code into a machine code file, which is then executed.
4. An _____ reads one statement from the source code, translates it to machine code or virtual machine code, and then executes it immediately.
5. The _____ command from the Software Development Kit executes a Python application.
6. A Python program file must end with the _____ file extension.
7. Programming errors can be categorized into three types: _____, _____, and _____.

Answer: 1. bit, 2. byte, 3. compiler, 4. interpreter, 5. python, 6. .py, 7. syntax errors, runtime errors, logic errors.

Further Reading

- Read Sections 1.2 - 1.3 and 1.5 - 1.8 of “Introduction to Python Programming and Data Structures” textbook.
- Read the guide about how to download and install Python Software Development Kit and Visual Studio Code:

https://course.cse.ust.hk/comp1023_new/python-vscode/index.html



That's all!

Any questions?

