Lecture slides - Week 13

Python - Exception Handling

Dr. Aamir Akbar

Director of both AWKUM AI Lab and AWKUM Robotics, Final Year Projects (FYPs) coordinator, and lecturer at the department of Computer Science
Abdul Wali Khan University, Mardan (AWKUM)

Contents

Exception Handling

Compile time and run-time errors

Compile time error is also called syntax error. These are errors that occur when the structure of the code is incorrect. For example, in Python if you forget to close a parenthesis, use a variable that is not defined, or indentation error (if you forget to indent your code correctly) are all syntax error.

```
if x > y:
a = 2    # indentation error
```

Runtime error occur when the code is executed. For example, if you try to access an element in a list with an index that is out of bounds, you will get a runtime error.

```
my_list = [1,2,3]
index = int(input("Enter index: "))
print(my_list[index])    # e.g., IndexError if user enters 3
```

Python Exception handling: try, except, finally i

In Python, `try` and `except` statements are used to handle exceptions. Also, `finally` block can be used to execute code regardless of whether an exception was raised or not. For example:

```
try:

    a = int(input("Enter first number: "))
    b = int(input("Enter second number: "))
    print(f"{a} / {b} = {a/b}")

except:
    print("A divide-by-zero error occurred")

finally:
    print("See you :)")
```

Python Exception handling: try, except, finally ii

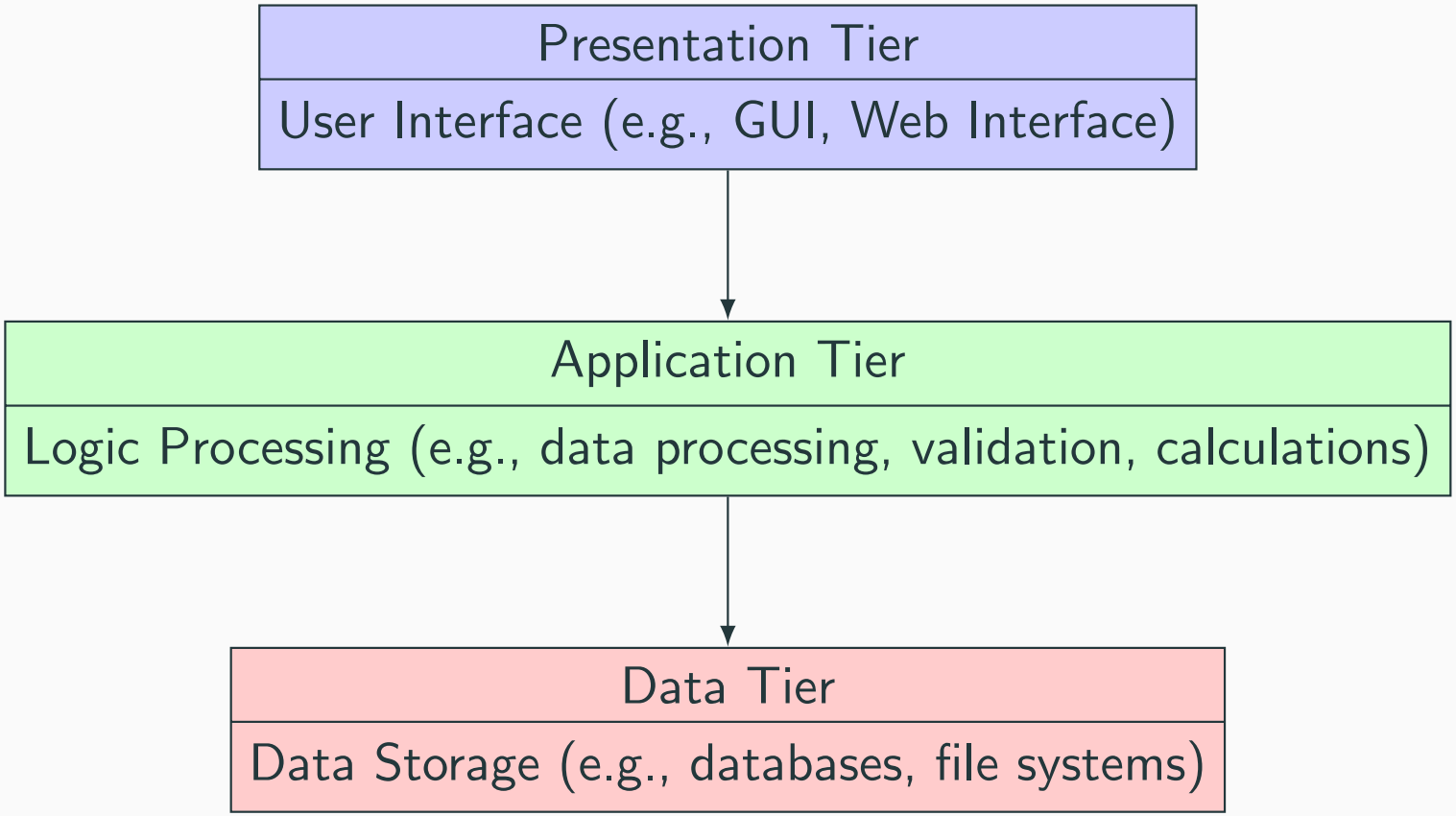
```
try:
    file = open("example.txt", "r")  # Opening a file in read mode
    # Perform operations on the file
    for line in file:
        print(line.strip())

except FileNotFoundError:
    print("File not found.")

except:
    print("An error occurred while processing the file.")

finally:
    if 'file' in locals():  # Checking if the file was successfully opened
        file.close()  # Closing the file whether an exception occurred or not
        print("File closed.")

print("Done with file operations.")
```

The `finally` block ensures that the file is closed, even if an exception occurs during the file operations or if the file is not found.

Case Study: Password Manager (3-tier Application Development)

What is 3-tier architecture in App development? i

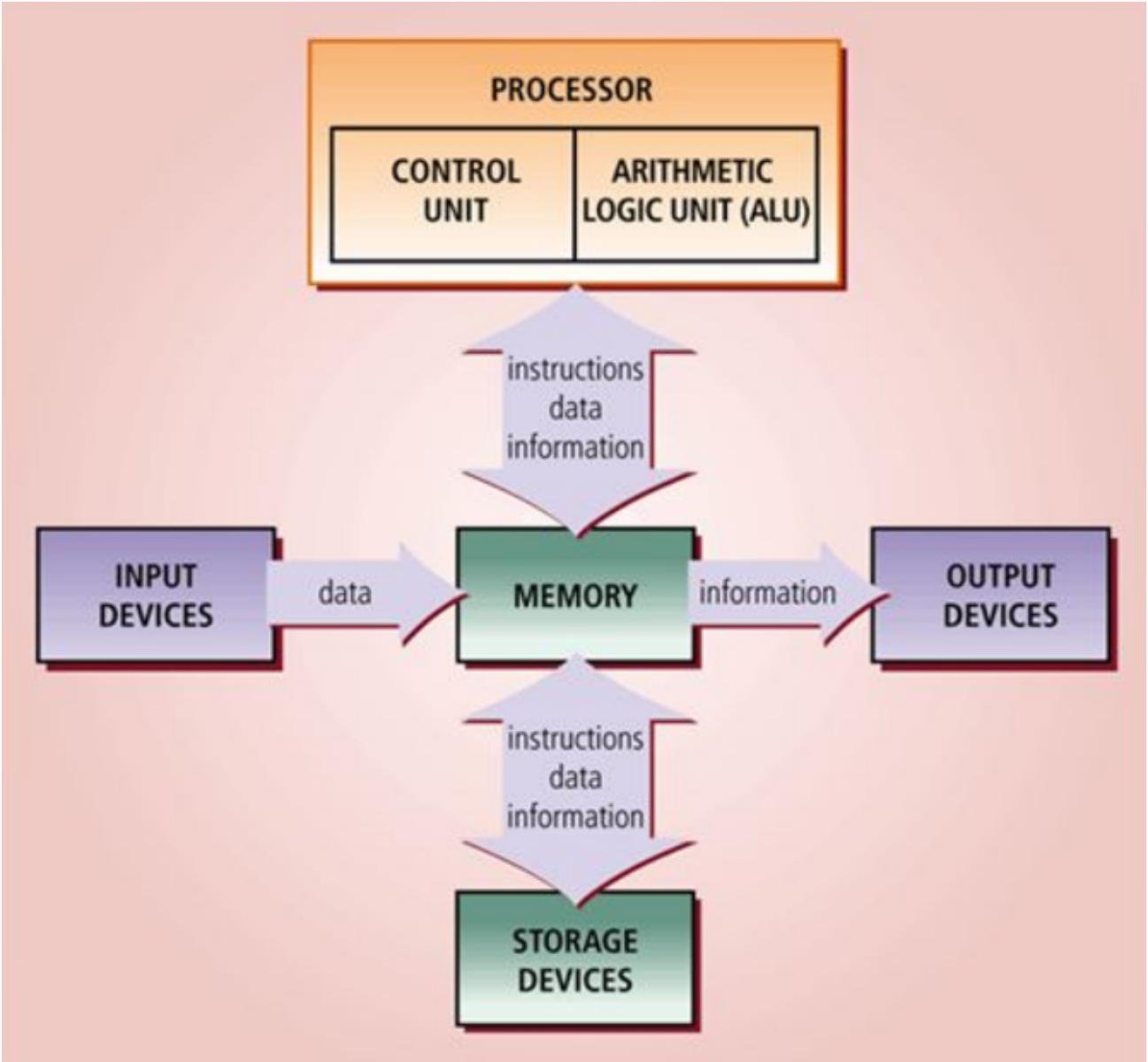
The `3-tier architecture` is a popular model used in software development to create scalable and modular applications.



What is 3-tier architecture in App development? ii

1. **Presentation Tier (User Interface):** This is the topmost layer that interacts directly with users. It's responsible for presenting information to users and collecting user inputs. It includes GUI components, web interfaces, or any other means through which users interact with the application.
2. **Application Tier (Logic Processing):** Also known as the business logic or middle tier. It contains the logic and processes that manipulate and handle data based on user inputs from the presentation tier. Here, data processing, validation, calculations, and other business-specific rules are implemented.
3. **Data Tier (Data Storage):** This layer is responsible for storing and managing data. It includes databases, file systems, or any other data storage systems. It stores the actual data required by the application and provides mechanisms to access and manipulate that data.

Your Application and Hardware Interaction



Bottom-up approach

What is the bottom-up approach in software application development process?

In a 3-tier architecture, a bottom-up approach refers to the development process that starts from the lowest level of the architecture, the Data Tier, and progresses upwards through the Application Tier to the Presentation Tier.

- Development begins by designing and implementing the data storage and management system.
- Once the Data Tier is established, the development moves to the Application Tier. This layer focuses on implementing the business logic, processing data obtained from the Data Tier, and performing various operations required by the application. Developers work on functionalities, algorithms, and rules that manipulate the data received from the Data Tier to meet the application's requirements.
- After the Application Tier is functional, the development progresses to the top layer, the Presentation Tier. Here, the focus is on

Data Tier

Activity: In our application development process, we aim to store account passwords permanently. Find what would be the appropriate data structure for storing this data and how to store this data permanently.

Data Tier

Activity: In our application development process, we aim to store account passwords permanently. Find what would be the appropriate data structure for storing this data and how to store this data permanently.

Data structure to use: In Python, we can use dictionary to map passwords to the provided accounts.

How to permanently store: Python dictionary is a JSON like structure, therefore, we can store this data in a JSON file system.

Application Tier

Activity: In the Application Tier, your task is to implement the core functionalities of the password manager. - Implement methods for securely hashing passwords using SHA-256 encryption.

The PasswordManager class serves as the foundation for securely managing passwords. It includes essential methods to set and verify passwords securely. When a user sets a password, the class hashes the provided password and stores it securely in a JSON file. Additionally, it provides a mechanism to verify if a user-provided password matches the stored hashed password for a specific account.

- Implement methods for securely hashing passwords using SHA-256 encryption.

- Develop functionalities to store and retrieve hashed passwords in/from a JSON file.

- Verify password correctness by comparing user-provided passwords with the stored hashed passwords.

Presentation Tier

Activity: The task is to create a Secure Password Manager GUI.

The GUI elements include entry fields for entering the account name and password, buttons to set and check passwords, and a label to display the result of password operations.

The standard Graphical User Interface (GUI) library for Python is `tkinter` (short for TK Interface). It provides a simple and intuitive way to create graphical interfaces and windows for desktop applications.

Import and use the necessary GUI module `tkinter`, and the `PasswordManager` class that we have created.