



**The German University in Cairo (GUC)**  
**Faculty of Media Engineering and Technology**  
**Computer Science and Engineering**  
**Computer System Architecture - CSEN 601**

---

## **Milestone 1 Project CA**

---

**Mohamed Ibrahim Mohamed Ahmed**

*Under Supervision of:*  
**Dr. Eng. Catherine M. Elias**

**Spring 2025**

# Contents

|          |                           |          |
|----------|---------------------------|----------|
| <b>1</b> | <b>Project Objectives</b> | <b>1</b> |
| <b>2</b> | <b>Introduction</b>       | <b>2</b> |
| <b>3</b> | <b>Methodology</b>        | <b>3</b> |
| <b>4</b> | <b>Results</b>            | <b>4</b> |
| <b>5</b> | <b>Conclusion</b>         | <b>5</b> |

# Chapter 1

## Project Objectives

The primary objective of Milestone 1 is to familiarize students with **C programming** and setting up a proper **coding environment** for file handling. The milestone focuses on creating a **C program** that reads a `.txt` file and prints its contents **line by line** to the terminal. This project also aims to:

- Develop **basic file handling skills** in C.
- Enhance students' understanding of **parsing text files**.
- Implement proper **error handling mechanisms**.
- Prepare students for future milestones involving **complex system interactions**.

## Chapter 2

# Introduction

File handling is an essential skill in **Computer Architecture**, as many real-world applications involve reading from and writing to files. This project serves as an introduction to:

- **Opening and reading files** using `fopen()`.
- **Parsing file contents line by line** with `fgets()`.
- **Handling errors** when a file cannot be opened.
- **Compiling and executing C programs** in a Linux/Windows environment.

In this milestone, the team developed a **C program** that reads a `.txt` file and displays its content in the terminal. Each line is parsed and printed separately to ensure **structured output**.

## Chapter 3

# Methodology

The project was implemented using the **C programming language**. Below are the key steps followed:

### Step 1: Setting Up the Environment

- Installed **GCC Compiler** (MinGW for Windows or GCC for Linux/macOS).
- Used **Visual Studio Code** as the preferred IDE.

### Step 2: Writing the C Code

A **C program** was developed to:

1. **Open a file** (`input.txt`) in **read mode**.
2. **Read each line separately** using `fgets()`.
3. **Print each line** with a corresponding line number.
4. **Handle file errors** if the file does not exist.
5. **Pause the execution** before closing to allow review of output.

### Step 3: Compilation & Execution

**For Windows (MinGW Required):**

```
gcc -o read_file.exe read_file.c  
read_file.exe
```

### Step 4: Testing the Program

## Chapter 4

### Results

The developed C program successfully:

**Opened and read a .txt file.**

**Printed each line separately.**

**Numbered the lines correctly.**

**Handled file errors properly.**

**Maintained a clear structure for easy debugging.**

The project met all functional requirements and worked as expected.

## Chapter 5

# Conclusion

This milestone helped students gain hands-on experience with **file handling in C**. The key takeaways include:

- Understanding the **importance of structured programming**.
- Learning how to **open, read, and parse files**.
- Gaining proficiency in **error handling**.
- Developing good **debugging practices**.