**แบบฝึกหัดเกี่ยวกับการจัดการไฟล์ใน C++ (ifstream, ofstream, fstream)**

**วัตถุประสงค์ของแบบฝึกหัด**

1. เข้าใจหลักการ **อ่านและเขียนไฟล์** ใน C++
2. ฝึกใช้ ofstream, ifstream, fstream เพื่อจัดการไฟล์
3. ใช้ **โหมดการเปิดไฟล์** (ios::in, ios::out, ios::app, ios::trunc)
4. เข้าใจ **พาธไฟล์** และ **การใช้โฟลเดอร์**
5. **🚀 สรุปแบบฝึกหัด**

| **แบบฝึกหัด** | **แนวคิดหลัก** | **ไฟล์ที่เกี่ยวข้อง** |
| --- | --- | --- |
| **1**: เขียนและอ่านไฟล์ข้อความ | ofstream, ifstream, getline() | message.txt |
| **2**: คำนวณและบันทึกผลลัพธ์ | ofstream, ifstream, คำนวณทางคณิตศาสตร์ | calculation.txt |
| **3**: อ่านข้อมูล CSV | ifstream, getline(), การแยกข้อมูล (,) | data.csv |
| **4**: บันทึก Log การใช้งาน | ofstream (ios::app), time() | log.txt |
| **5**: อ่านค่า Config | ifstream, getline(), การจัดการ key=value | config.txt |
| **6**: แก้ไขข้อมูลในไฟล์ | fstream, ifstream, ofstream | data.txt |

|  |  |  |
| --- | --- | --- |
| **ฟังก์ชัน** | **C++** | **Python** |
| **เขียนและอ่านไฟล์ข้อความ** | ofstream, ifstream | open("file.txt", "w/r") |
| **คำนวณและบันทึกผลลัพธ์** | ofstream, ifstream | open("file.txt", "w/r") |
| **อ่านไฟล์ CSV** | ifstream, getline() | csv.reader() |
| **บันทึก Log การใช้งาน** | ofstream (ios::app), time() | datetime.now() |
| **อ่านค่า Config** | ifstream, getline() | open("config.txt", "r") |
| **แก้ไขข้อมูลในไฟล์** | fstream, ofstream | file.read/write() |

**🔹 แบบฝึกหัด 1: เขียนและอ่านไฟล์ข้อความ (.txt)**

**ดูตัวอย่าง code ที่ LAB12\_06**

**📌 โจทย์**

1. ให้เขียนโปรแกรมรับข้อความจากผู้ใช้
2. บันทึกข้อความลงในไฟล์ message.txt
3. จากนั้นเปิดไฟล์ขึ้นมาอ่าน แล้วแสดงผลข้อความทั้งหมดที่ถูกบันทึก
4. หากไฟล์ยังไม่มี ให้แจ้งเตือนว่า "ไฟล์ยังไม่มีข้อมูล"

**📌 ตัวอย่างผลลัพธ์ที่ต้องการ**

|  |
| --- |
| Enter your message: Hello, this is a test message!  [บันทึกข้อมูลเรียบร้อย!]  [อ่านข้อมูลจากไฟล์ message.txt]  Hello, this is a test message! |

**Code C++**

|  |
| --- |
| #include <iostream>  #include <fstream>  #include <string>  using namespace std;  int main() {  string message;  cout << "Enter your message: ";  getline(cin, message);  ofstream outFile("message.txt");  outFile << message << endl;  outFile.close();  ifstream inFile("message.txt");  if (inFile) {  string line;  cout << "Contents of message.txt:" << endl;  while (getline(inFile, line)) {  cout << line << endl;  }  inFile.close();  } else {  cout << "The file has no data." << endl;  }  return 0;  } |

**ผลการทดลอง**

|  |
| --- |
|  |

**Code Python**

|  |
| --- |
| message = input("Enter your message: ")  with open("message.txt", "w") as outFile:  outFile.write(message + "\n")  try:  with open("message.txt", "r") as inFile:  print("Contents of message.txt:")  for line in inFile:  print(line, end="")  except FileNotFoundError:  print("The file has no data.") |

**ผลการทดลอง**

|  |
| --- |
|  |

**🔹 แบบฝึกหัด 2: คำนวณและบันทึกผลลัพธ์ลงไฟล์**

**ดูตัวอย่าง code ที่ LAB12\_07**

**โจทย์**

1. ให้โปรแกรมรับตัวเลข 2 ค่า (a และ b) จากผู้ใช้
2. คำนวณผลลัพธ์ของ a + b, a - b, a \* b, a / b
3. บันทึกผลลัพธ์ลงไฟล์ calculation.txt
4. เปิดไฟล์ calculation.txt แล้วแสดงผลข้อมูลที่บันทึกไว้

**ตัวอย่างผลลัพธ์ที่ต้องการ**

|  |
| --- |
| Enter first number: 10  Enter second number: 5  [บันทึกผลลัพธ์เรียบร้อย!]  [อ่านข้อมูลจากไฟล์ calculation.txt]  10 + 5 = 15  10 - 5 = 5  10 \* 5 = 50  10 / 5 = 2 |

**Code**

|  |
| --- |
| #include <iostream>  #include <fstream>  using namespace std;  int main() {  double a, b;  cout << "Enter first number: ";  cin >> a;  cout << "Enter second number: ";  cin >> b;  double sum = a + b;  double diff = a - b;  double prod = a \* b;  double quot = a / b;  ofstream outfile("calculation.txt");  outfile << a << " + " << b << " = " << sum << endl;  outfile << a << " - " << b << " = " << diff << endl;  outfile << a << " \* " << b << " = " << prod << endl;  outfile << a << " / " << b << " = " << quot << endl;  outfile.close();  cout << "[บันทึกผลลัพธ์เรียบร้อย!]" << endl;  ifstream infile("calculation.txt");  string line;  cout << "\n[อ่านข้อมูลจากไฟล์ calculation.txt]" << endl;  while (getline(infile, line)) {  cout << line << endl;  }  infile.close();  return 0;  } |

**ผลการทดลอง**

|  |
| --- |
|  |

**Code Python**

|  |
| --- |
| def main():  a = float(input("Enter first number: "))  b = float(input("Enter second number: "))  sum\_result = a + b  diff\_result = a - b  prod\_result = a \* b  quot\_result = a / b  with open("calculation.txt", "w") as file:  file.write(f"{a} + {b} = {sum\_result}\n")  file.write(f"{a} - {b} = {diff\_result}\n")  file.write(f"{a} \* {b} = {prod\_result}\n")  file.write(f"{a} / {b} = {quot\_result}\n")  print("[บันทึกผลลัพธ์เรียบร้อย!]")  print("\n[อ่านข้อมูลจากไฟล์ calculation.txt]")  with open("calculation.txt", "r") as file:  for line in file:  print(line.strip())  if \_\_name\_\_ == "\_\_main\_\_":  main() |

**ผลการทดลอง**

|  |
| --- |
|  |

**🔹 แบบฝึกหัด 3: อ่านไฟล์ CSV และแสดงผลข้อมูล**

**ดูตัวอย่าง code ที่ LAB12\_08**

**📌 โจทย์**

1. ให้มีไฟล์ data.csv ที่เก็บข้อมูลพนักงานในรูปแบบ CSV เช่น:
2. ID,Name,Salary
3. 101,John,50000
4. 102,Alice,60000
5. 103,Bob,55000
6. ให้เขียนโปรแกรม **อ่านข้อมูลจากไฟล์ data.csv**
7. แสดงข้อมูลทั้งหมดออกมาทางหน้าจอในรูปแบบตาราง

**📌 ตัวอย่างผลลัพธ์ที่ต้องการ**

|  |
| --- |
| [แสดงข้อมูลจากไฟล์ data.csv]  ---------------------------------  | ID | Name | Salary |  ---------------------------------  | 101 | John | 50000 |  | 102 | Alice | 60000 |  | 103 | Bob | 55000 |  --------------------------------- |

**Code**

|  |
| --- |
| #include <iostream>  #include <fstream>  #include <sstream>  #include <vector>  #include <iomanip>  #include <stdexcept>  using namespace std;  struct Employee {  int id;  string name;  int salary;  };  int main() {  ifstream file("data.csv");  if (!file) {  cerr << "Unable to open file data.csv" << endl;  return 1;  }  string line;  vector<Employee> employees;  // Skip the header line  getline(file, line);  // Read the data  while (getline(file, line)) {  stringstream ss(line);  string id, name, salary;  getline(ss, id, ',');  getline(ss, name, ',');  getline(ss, salary, ',');  try {  int emp\_id = stoi(id);  int emp\_salary = stoi(salary);  employees.push\_back({emp\_id, name, emp\_salary});  } catch (const invalid\_argument& e) {  cerr << "Invalid data: " << line << endl;  } catch (const out\_of\_range& e) {  cerr << "Out of range data: " << line << endl;  }  }  file.close();  // Display the data in table format  cout << "---------------------------------" << endl;  cout << "| ID | Name | Salary |" << endl;  cout << "---------------------------------" << endl;  for (const auto& emp : employees) {  cout << "| " << setw(4) << emp.id << " | " << setw(6) << emp.name << " | " << setw(10) << emp.salary << " |" << endl;  }  cout << "---------------------------------" << endl;  return 0;  } |

**ผลการทดลอง**

|  |
| --- |
|  |

**Code Python**

|  |
| --- |
| import csv  employees = []  # Read the data from the CSV file  with open('data.csv', newline='') as csvfile:  reader = csv.DictReader(csvfile)  for row in reader:  employees.append(row)  # Display the data in table format  print("---------------------------------")  print("| ID | Name | Salary |")  print("---------------------------------")  for emp in employees:  print(f"| {emp['ID']:4} | {emp['Name']:6} | {emp['Salary']:10} |")  print("---------------------------------") |

**ผลการทดลอง**

|  |
| --- |
|  |

**🔹 แบบฝึกหัด 4: บันทึก Log การใช้งานลงไฟล์**

**📌 โจทย์**

1. ให้เขียนโปรแกรมที่บันทึก "กิจกรรมการใช้งาน" ลงไฟล์ log.txt
2. ทุกครั้งที่โปรแกรมทำงาน ให้เพิ่มข้อความ "User started the program at [timestamp]"
3. อ่านไฟล์ log.txt และแสดงผลประวัติการใช้งานทั้งหมด

**📌 ตัวอย่างผลลัพธ์ที่ต้องการ**

|  |
| --- |
| [บันทึกการใช้งานลงไฟล์เรียบร้อย]  [อ่านข้อมูลจาก log.txt]  User started the program at 2025-02-14 12:34:56  User started the program at 2025-02-14 12:45:20 |

**Code**

|  |
| --- |
| #include <iostream>  #include <fstream>  #include <ctime>  using namespace std;  void logActivity() {  ofstream logFile("log.txt", ios\_base::app);  if (logFile.is\_open()) {  time\_t now = time(nullptr);  logFile << "User started the program at " << ctime(&now);  logFile.close();  cout << "[บันทึกการใช้งานลงไฟล์เรียบร้อย]\n";  } else {  cerr << "Unable to open log file.\n";  }  }  void readLog() {  ifstream logFile("log.txt");  if (logFile.is\_open()) {  cout << "\n[อ่านข้อมูลจาก log.txt]\n";  string line;  while (getline(logFile, line)) {  cout << line << "\n";  }  logFile.close();  } else {  cerr << "Unable to open log file.\n";  }  }  int main() {  logActivity();  readLog();  return 0;  } |

**ผลการทดลอง**

|  |
| --- |
|  |

**Code Python**

|  |
| --- |
| import datetime  def log\_activity():  with open("log.txt", "a") as log\_file:  now = datetime.datetime.now()  log\_file.write(f"User started the program at {now.strftime('%Y-%m-%d %H:%M:%S')}\n")  print("[บันทึกการใช้งานลงไฟล์เรียบร้อย]")  def read\_log():  try:  with open("log.txt", "r") as log\_file:  print("\n[อ่านข้อมูลจาก log.txt]")  for line in log\_file:  print(line, end="")  except FileNotFoundError:  print("Unable to open log file.")  if \_\_name\_\_ == "\_\_main\_\_":  log\_activity()  read\_log() |

**ผลการทดลอง**

|  |
| --- |
|  |

**🔹 แบบฝึกหัด 5: อ่านค่า Config จากไฟล์**

**📌 โจทย์**

1. ให้มีไฟล์ config.txt ที่มีค่าตั้งค่า เช่น:
2. speed=2.5
3. max\_distance=150
4. mode=auto
5. ให้เขียนโปรแกรม **อ่านค่า config จากไฟล์**
6. แสดงค่าที่อ่านได้ออกมาทางหน้าจอ

**📌 ตัวอย่างผลลัพธ์ที่ต้องการ**

|  |
| --- |
| [Robot Configuration]  Speed: 2.5 m/s  Max Distance: 150 cm  Mode: Auto |

**Code**

|  |
| --- |
| #include <iostream>  #include <fstream>  #include <string>  #include <map>  using namespace std;  map<string, string> read\_config(const string& file\_path) {  map<string, string> config;  ifstream file(file\_path);  string line;  while (getline(file, line)) {  size\_t pos = line.find('=');  if (pos != string::npos) {  string key = line.substr(0, pos);  string value = line.substr(pos + 1);  config[key] = value;  }  }  return config;  }  void display\_config(const map<string, string>& config) {  cout << "[Robot Configuration]" << endl;  cout << "Speed: " << config.at("speed") << " m/s" << endl;  cout << "Max Distance: " << config.at("max\_distance") << " cm" << endl;  cout << "Mode: " << config.at("mode") << endl;  }  int main() {  map<string, string> config = read\_config("config.txt");  display\_config(config);  return 0;  } |

**ผลการทดลอง**

|  |
| --- |
|  |

**Code Python**

|  |
| --- |
| def read\_config(file\_path):  config = {}  with open(file\_path, 'r') as file:  for line in file:  key, value = line.strip().split('=')  config[key] = value  return config  def display\_config(config):  print("[Robot Configuration]")  print(f"Speed: {config['speed']} m/s")  print(f"Max Distance: {config['max\_distance']} cm")  print(f"Mode: {config['mode'].capitalize()}")  if \_\_name\_\_ == "\_\_main\_\_":  config = read\_config('config.txt')  display\_config(config) |

**ผลการทดลอง**

|  |
| --- |
|  |

**🔹 แบบฝึกหัด 6: แก้ไขข้อมูลในไฟล์**

**ดูตัวอย่าง code ที่ LAB12\_09**

**📌โจทย์**

1. ให้มีไฟล์ data.txt ที่มีข้อมูลเดิม เช่น:
2. Apple
3. Banana
4. Cherry
5. ให้เขียนโปรแกรมที่สามารถ **แก้ไขข้อมูลในไฟล์** ได้
6. ให้ผู้ใช้ป้อน **"คำที่ต้องการแก้ไข" และ "คำใหม่ที่จะแทนที่"**
7. บันทึกข้อมูลใหม่ลงไฟล์

**📌 ตัวอย่างผลลัพธ์ที่ต้องการ**

|  |
| --- |
| [ไฟล์เดิม: data.txt]  Apple  Banana  Cherry  Enter word to replace: Banana  Enter new word: Mango  [แก้ไขข้อมูลในไฟล์เรียบร้อย!]  [ไฟล์ใหม่: data.txt]  Apple  Mango  Cherry |

**Code**

|  |
| --- |
| #include <iostream>  #include <fstream>  #include <string>  #include <vector>  using namespace std;  int main() {  ifstream file("data.txt");  vector<string> lines;  if (!file) {  cerr << "Unable to open file data.txt. Creating a new file with default values." << endl;  vector<string> default\_data = {"Hello", "World", "C++", "first initial"};  ofstream outfile("data.txt");  for (const auto &l : default\_data) {  outfile << l << endl;  }  outfile.close();  lines = default\_data;  } else {  string line;  while (getline(file, line)) {  lines.push\_back(line);  }  file.close();  }  string old\_word, new\_word;  cout << "Enter word to replace: ";  cin >> old\_word;  cout << "Enter new word: ";  cin >> new\_word;  for (auto &l : lines) {  if (l == old\_word) {  l = new\_word;  }  }  ofstream outfile("data.txt");  for (const auto &l : lines) {  outfile << l << endl;  }  outfile.close();  cout << "[แก้ไขข้อมูลในไฟล์เรียบร้อย!]" << endl;  return 0;  } |

**ผลการทดลอง**

|  |
| --- |
|  |

**Code Python**

|  |
| --- |
| def replace\_word\_in\_file(file\_path, old\_word, new\_word):  with open(file\_path, 'r') as file:  lines = file.readlines()  with open(file\_path, 'w') as file:  for line in lines:  if line.strip() == old\_word:  file.write(new\_word + '\n')  else:  file.write(line)  if \_\_name\_\_ == "\_\_main\_\_":  file\_path = 'data.txt'  old\_word = input("Enter word to replace: ")  new\_word = input("Enter new word: ")  replace\_word\_in\_file(file\_path, old\_word, new\_word)  print("[แก้ไขข้อมูลในไฟล์เรียบร้อย!]") |

**ผลการทดลอง**

|  |
| --- |
|  |