

OBJECTIVE

- To understand the concept and importance of file handling in C++ programming.
- To learn how to use the file stream classes: ifstream, ofstream, and fstream.

BACKGROUND THEORY

File handling in C++ is a crucial feature that enables programs to store data permanently on disk files, allowing data to persist beyond the life of a program's execution. Unlike variables stored in memory, which are lost once a program terminates, files provide a way to save and retrieve information for later use. This is essential for many real-world applications such as databases, logging systems, configuration files, and more.

C++ supports file handling through the <fstream> library, which provides three primary classes: ifstream for input file streams (reading from files), ofstream for output file streams (writing to files), and fstream for both reading and writing. These classes allow programs to open files, read data from them, write data to them, and close them when finished. Files can be opened in different modes, such as read-only, write-only, append mode, or binary mode, depending on the requirements of the application.

When opening a file, different modes can be specified:

- ios::in – Read mode
- ios::out – Write mode
- ios::app – Append to end of file
- ios::ate – Go to end of file on opening
- ios::trunc – Truncate file if it exists
- ios::binary – Open in binary mode

Write a C++ program using structure and file handling to input details of multiple students and save them to a file.

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <vector>
```

```
#include <string>
```

```
using namespace std;
```

```
struct Student {
```

```
int roll;
string name;
int age;
string course;
};

int main() {
    int n;
    cout << "Enter number of students: ";
    cin >> n;
    cin.ignore();
    vector<Student> students;
    for (int i = 0; i < n; ++i) {
        Student s;
        cout << "\nEnter details for student " << i + 1 << ":\n";
        cout << "Roll number: ";
        cin >> s.roll;
        cin.ignore();
        cout << "Name: ";
        getline(cin, s.name);
        cout << "Age: ";
        cin >> s.age;
        cin.ignore();
        cout << "Course: ";
        getline(cin, s.course);
        students.push_back(s);
    }

    ofstream fout("students.txt");
```

```

if (!fout) {
    cout << "Error opening file for writing.\n";
    return 1;
}
for (const auto& s : students) {
    fout << s.roll << "," << s.name << "," << s.age << "," << s.course << endl;
}
fout.close();

cout << "\n--- Student Details ---\n";
for (const auto& s : students) {
    cout << "Roll: " << s.roll << endl;
    cout << "Name: " << s.name << endl;
    cout << "Age: " << s.age << endl;
    cout << "Course: " << s.course << endl;
    cout << "-----\n";
}
return 0;
}

```

Write a C++ program using structure and file handling to store and display inventory item details.

```

#include <iostream>
#include <vector>
#include <string>
#include <fstream>
using namespace std;

struct Item {

```

```
int item_ID;
string name;
double price;
string mfd_date;
string company;
};

int main() {
    int n;
    cout << "Enter number of items: ";
    cin >> n;
    cin.ignore();

    vector<Item> inventory;

    for (int i = 0; i < n; ++i) {
        Item temp;
        cout << "\nEnter details for item " << i + 1 << ":\n";
        cout << "Item ID: ";
        cin >> temp.item_ID;
        cin.ignore();

        cout << "Name: ";
        getline(cin, temp.name);

        cout << "Price: ";
        cin >> temp.price;
        cin.ignore();
```

```

    cout << "Manufacturing Date (YYYY-MM-DD): ";
    getline(cin, temp.mfd_date);

    cout << "Company: ";
    getline(cin, temp.company);

    inventory.push_back(temp);
}

ofstream fout("inventory.txt");
if (!fout) {
    cout << "Error opening file for writing.\n";
    return 1;
}

for (const auto& item : inventory) {
    fout << item.item_ID << "," << item.name << "," << item.price << "," << item.mfd_date <<
    "," << item.company << endl;
}

fout.close();

cout << "\n--- Inventory Records ---\n";
for (const auto& item : inventory) {
    cout << "Item ID: " << item.item_ID << endl;
    cout << "Name: " << item.name << endl;
    cout << "Price: $" << item.price << endl;
    cout << "Manufacturing Date: " << item.mfd_date << endl;
    cout << "Company: " << item.company << endl;
    cout << "-----\n";
}

```

```
}

cout << "Inventory has been saved to inventory.txt\n";

return 0;
}
```

Discussion

In this lab, we explored the concept of file handling in C++ and how it enables programs to store and retrieve data from external files. By using the `<fstream>` library and its classes (`ifstream`, `ofstream`, and `fstream`), we were able to create programs that write structured data into a file and read it back when needed. Specifically, we implemented a program that collects item details such as ID, name, price, manufacturing date, and company, stores them in a structure, and saves the data in a text file in a comma-separated format.

Conclusion

The lab on file handling provided practical experience in managing data input/output through external files using C++. We learned how to use structures along with file streams to organize and store data efficiently. The ability to write data into a file and read it back is fundamental for many applications such as inventory systems, student records, and log files. Overall, this lab enhanced our understanding of how file operations can be used to develop more robust and persistent software solutions.