Khaula Molapo

C++ Week 10

1. Personal Diary Program

This program allows you to write and read diary entries stored in a file.

```
#include <iostream>
#include <fstream>
#include <string>
#include <ctime>
using namespace std;

int main() {
   int choice;
   string entry;
   ifstream readFile;
   ofstream writeFile;

   cout << "==== PERSONAL DIARY ====
";
   cout << "1. Add new entry
";
   cout << "2. Read previous entries
";
   cout << "Enter your choice: ";
   cin >> choice;
   cin.ignore();

   if (choice == 1) {
      writeFile.open("diary.txt", ios::app);
      if (!writeFile) {
         cout << "Error opening file!
";
         return 1;
      }

      cout << "Write your diary entry: ";
      getline(cin, entry);

      time_t now = time(0);
      string dt = ctime(&now);

      writeFile << "---- " << dt << " ----
```

```
";
        writeFile << entry << "\n\n";
        writeFile.close();
        cout << "Entry saved!
";
    }

    else if (choice == 2) {
        readFile.open("diary.txt");
        if (!readFile) {
            cout << "No diary found yet.\n";
            return 1;
        }

        cout << "\n=== YOUR DIARY ENTRIES ===\n";
        string line;
        while (getline(readFile, line)) {
            cout << line << endl;
        }
        readFile.close();
    }

    else {
        cout << "Invalid choice!\n";
    }

    return 0;
}
```

2. CSV File Parser

This program reads data from a CSV file, displays it, and writes it to a new file in formatted form.

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
using namespace std;

int main() {
    ifstream inputFile("data.csv");
    ofstream outputFile("formatted_data.txt");

    if (!inputFile) {
```

```
    cout << "Error: data.csv not found!" << endl;
    return 1;
  }

  string line;
  cout << "=== CSV Data ===\n";
  while (getline(inputFile, line)) {
    stringstream ss(line);
    string name, age, job;

    getline(ss, name, ',');
    getline(ss, age, ',');
    getline(ss, job, ',');

    cout << "Name: " << name << ", Age: " << age << ", Job: " << job << endl;
    outputFile << "Name: " << name << " | Age: " << age << " | Job: " << job << "\n";
  }

  inputFile.close();
  outputFile.close();
  cout << "\nData saved to 'formatted_data.txt'\n";
  return 0;
}
```

3. Object Persistence - Book Collection

This program stores a list of books in a file, allows adding new books, and loads them when restarted.

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

class Book {
public:
  string title;
  string author;
  int year;

  void input() {
    cout << "Enter title: ";
    getline(cin, title);
    cout << "Enter author: ";
```

```
        getline(cin, author);
        cout << "Enter year: ";
        cin >> year;
        cin.ignore();
    }

    void display() {
        cout << "Title: " << title << ", Author: " << author << ", Year: " << year << endl;
    }
};

void saveBooks(vector<Book>& books) {
    ofstream file("books.txt");
    for (auto &b : books) {
        file << b.title << "," << b.author << "," << b.year << "\n";
    }
    file.close();
}

void loadBooks(vector<Book>& books) {
    ifstream file("books.txt");
    if (!file) return;
    string line;
    while (getline(file, line)) {
        Book b;
        size_t pos1 = line.find(',');
        size_t pos2 = line.rfind(',');
        b.title = line.substr(0, pos1);
        b.author = line.substr(pos1 + 1, pos2 - pos1 - 1);
        b.year = stoi(line.substr(pos2 + 1));
        books.push_back(b);
    }
    file.close();
}

int main() {
    vector<Book> books;
    loadBooks(books);
    int choice;

    cout << "==== BOOK COLLECTION ====" << endl;
    cout << "1. Add new book" << endl;
    cout << "2. Show all books" << endl;
```

```
    cout << "Enter your choice: ";
    cin >> choice;
    cin.ignore();

    if (choice == 1) {
        Book b;
        b.input();
        books.push_back(b);
        saveBooks(books);
        cout << "Book saved!" << endl;
    }
    else if (choice == 2) {
        if (books.empty()) cout << "No books found." << endl;
        else for (auto &b : books) b.display();
    }
    else {
        cout << "Invalid choice!" << endl;
    }

    return 0;
}
```