**Personal Finance Management Console Application using C++ OOP**

*By: Usaid Rehan (20K-0297) and Huzaifa Jawwad (20K-0175)*

**Problem Statement:**

In the age of thriftless spending, mismanagement of personal finances and lack of financial education, there is a need for a console application that will help users manage their personal finances responsibly.  
  
The application should display minimalist intro and outro screens at starting and ending respectively including the request of password to access the account. Their application should store the user’s bank balance, emergency fund, total revenue and total expenses in a time frame, as well as, whether the user is a tax filer or not. Such data should be encapsulated and not directly accessed or modified.

The account manages two main types of data: expenses and revenue. The expenses managed by the program should be categorized into recurring expenses and one-time expenses. The revenue managed by the program should be categorized into investment, income and passive earnings. The investment is further divided into two major types: Stocks and Real Estate.

The possibilities of adding components to these categories are limitless, for example, there can be separate data for Commercial and Industrial Real Estate while stocks data can be further divided into Individual Stocks, Index Funds, Mutual Funds or even IPOs and there will always be a possibility of adding unique investment opportunities such as precious metals like Gold or Silver. The passive earnings will comprise of dividend earnings, royalties, as well as, passive income from businesses.

The application should also incorporate functions inspired from Data Science such as Data Visualization, Data Analytics and Machine Learning, while building the application platform can be considered as close to implementing Data Engineering.

**Solution:**

The program will be implemented using object-oriented programming using C++ compiled in Dev C++. Classes will be made for the user account and each category. Inheritance will be based on the hierarchy and links of each category, for example, Investment class will inherit from both Revenue Class, as well as, One-Time Expense Class. Objects will be made for each class and will be used to access their data members and behaviours. Global functions will be made for Data Visualization, Retirement Predictor (inspired from Machine Learning) and Investment Advisor (Inspired from Data Analytics).

**Main Functionalities:**

Constructors and Destructors

Accessors and Mutators

Member Functions

void Data\_Visualization(Data Parameters): Visualizes the data, in form of bar chart display

void Retirement\_Prediction(Data Parameters): Based on the data and retirement amount, predicts the year

void Investment\_Advisor(Data Parameters): Based on the data, it advises what user should invest in

**Entities (Classes):**

Account

Expense

OneTime

Recurring

Revenue

Investment

Stock

RealEstate

Income

Passive