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

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

In the age of thriftless spending, mismanagement of personal finances and lack of financial education, we came up with the idea of a personal finance management productivity tool so that people can fulfill the responsibility of putting their finances in order.

Our aim was to combine all the in-depth concepts of OOP in C++ along with the essence of Data Science into a practical and real-life application, with appealing Introduction and Conclusion, as well as, minimalist Menu displays just like Smartphone applications and making the console window more lively and user-friendly with enriched colours without using any graphics libraries, GUI tools or custom header files.

Contains 10 Classes, 10 Menu Functions (includes 3 Data Science, 1 Filing and 1 Exception Handling) in about 1500 Lines. Covers the 4 Pillars of OOP including both the types of Polymorphism. Includes Operator & Data Member Overloading, Interfaces, Explicit Keyword, Deep Copy Constructor, Downcasting, Exception Library, Templates, Maps & sstream Library, Vectors and its library's functions, Iterators, Object Filing, Abstract Class and Pure Virtual Functions.

The Data Science essence includes Data Visualization as bar chart, Data Analysis as Investment Advisor and AI style Investment Advisor while the whole can be considered as Engineering a Data Platform.

The first 4 functions are for viewing and editing the finances. The next function, that is, Data Visualization, incorporates the Data Science field of the same name, by displaying the Cryptocurrency or Stock Tickers owned by the user, in the form of a bar chart for comparison in the portfolio. Moreover, the Investment Advisor function is based on the Data Analytics field of Data Science, since it analyzes the portfolio to recommend to users which type of asset to invest in next. Furthermore, the Retirement Predictor is an AI style predictor, using an algorithm and data entered by the user, to predict how many years until retirement. Lastly, the Summary Filing and Proof of E2E Encryption functions are simple implementation of object filing and exception handling respectively.

All C++ OOP Concepts covered:

**A. Encapsulation**

**B. Abstraction**

**C. Inheritance (Virtual Hybrid / Diamond)**

**1. Polymorphism**

**i. (Compile-Time)**

Overloading of Account Constructor + Operator Overloading (++)

**ii. (Run-Time) / Abstract Class / Pure Virtual Function**

Behaviour to Calculate Total Expenses

**2. Interface**

Revenue Class and Investment Class

**3. Explicit Keyword + Deep Copy Constructor**

To avoid user from stealing password

**4. Data Member Overriding**

Important Expense Variables Specified in Base Class and Overriden in Derived

**5. Operator Overloading (++)**

Increment Login Count

**6. Downcasting**

Derived PERSON Class Pointer Dynamic Casted with Base EXPENSE Class Pointer in  
 E2E Encryption Function

**7. Exception Handling (bad\_typeid)**

TypeID Error in End to End Encryption Function

**8. Generic Programming / Templates (2 Generic Types)**

Investment Advisor Function

**9. Standard Template Library (Vector + Iterator + Map)**

Iterator and Map in Data Analytics Function

Vector and its functions in Investment Advisor Function

**10. Filing (Writing Object Data)**

Monthly Summary stored in Monthly\_Report.txt

