**National University of Computer & Emerging Sciences**

**Karachi Campus**



**Smart Bank Management System**

**Object-Oriented Programming**

**Section: F**

**Group Members:**

**24k-0808 Muhammad Saim Noman**

**24k-0533 Huzaifa Nadeem**

**24k-0866 Syed Rafay Mehdi**

Project Proposal

* **Introduction**

We are developing a **Smart Bank Management System** using **Object-Oriented Programming (OOP) in C++**. This system is designed to digitally manage essential banking operations in a secure, efficient, and intelligent manner. It provides an interactive platform for customers and bank staff to perform and manage various banking activities with ease.

* **Existing System**

Modern banking platforms like internet banking apps and core banking systems offer features such as account management, secure logins, and transaction history. Our system follows a similar structure but is simplified and built using C++ OOP concepts for educational and development purposes.

* **Problem Statement**

Existing systems lack flexibility, are not fully object-oriented, and often miss detailed transaction tracking or user-friendly role-based access. Our system introduces a smart, modular design using C++ OOP with features like **secure multi-role login, automated interest calculation**, and **clear transaction history,** making it more efficient and easier to maintain.

**Proposed Solution**

We will implement new features using OOP principles in C++. Each feature will be handled through separate classes—for example, a User class for login, Account class for account operations, and Transaction class for tracking activities. Inheritance will manage different user roles, while polymorphism will allow flexible operations like deposits and withdrawals. Secure login and interest calculations will be added using encapsulation and functionoverloading.

* **Salient Features**
* User Account Management
* Create, update, delete accounts.
* Account types: Savings, Current.
* Transactions
* Deposit, Withdraw, Transfer between accounts.
* Transaction history stored using file I/O (binary or text files).
* Interest Calculation / Charges
* Use polymorphism: each account type calculates interest differently.

**Tools & Technologies**

**Programming Language**: C++

**Operating System**: Windows (compatible with Linux as well)

**Development Tools**: Code::Blocks / Dev C++ / Visual Studio