A blue circle with yellow and white text

AI-generated content may be incorrect.

**The University of Azad Jammu and Kashmir,**

**Muzaffarabad**

|  |  |
| --- | --- |
| Name | Kamal Ali Akmal |
| Course Name | Data Structure & Algorithm |
| Submitted to | Engr. Sidra Rafique |
| Semester | 3rd |
| Session | 2024-2028 |
| Roll No | 2024-SE-38 |
| Lab No | 03 |
| Submission date | 29 October 2025 |

# Abstract Data Type (ADT) – Bank Account System

## Objective:

To design and implement an Abstract Data Type (ADT) for a real-world application using C++. This program demonstrates abstraction and encapsulation through a simple Bank Account system.

## A screenshot of a computer code AI-generated content may be incorrect.Code:

**Output Example:**

A screenshot of a computer

AI-generated content may be incorrect.

## Explanation:

1. The class `BankAccount` represents the Abstract Data Type (ADT).

2. Private members (accountHolder, accountNumber, balance) store data internally.

3. Public functions (deposit, withdraw, checkBalance, displayAccount) allow safe interaction with data.

4. The implementation details are hidden, showing only the essential operations to the user.

## Operations Supported by the ADT:

• Deposit money – Adds amount to the account balance.

• Withdraw money – Subtracts amount if balance is sufficient.

• Check balance – Displays current available balance.

• Display account details – Shows account holder information and current balance.

## Demonstration of Abstraction and Encapsulation:

• Encapsulation: The data members are private and cannot be accessed directly outside the class. Only class methods can modify them, ensuring data protection.

• Abstraction: The user interacts through simple operations like deposit () and withdraw(), without knowing the internal implementation details.

## Conclusion:

The Bank Account ADT provides a clean and modular representation of a real-world system. It defines necessary operations while hiding implementation details, demonstrating the key principles of abstraction and encapsulation in the context of Data Structures and Object-Oriented Programming.