```cpp
#include <iostream>
#include <string>
using namespace std;

class BankAccount {
private:
    int accountNumber;
    double balance;
    string accountHolderName;

public:
    // Constructor
    BankAccount(int accNum, double initialBalance, string holderName)
        : accountNumber(accNum), balance(initialBalance), accountHolderName(holderName) {}

    // Function to deposit an amount
    void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            cout << "Deposited " << amount << " into account " << accountNumber << ".\n";
        } else {
            cout << "Invalid deposit amount.\n";
        }
    }

    // Function to withdraw an amount
    void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            cout << "Withdrew " << amount << " from account " << accountNumber << ".\n";
        } else {
            cout << "Insufficient balance or invalid amount for account " << accountNumber << ".\n";
        }
    }

    // Function to display balance
    void displayBalance() const {
        cout << "Account Number: " << accountNumber
             << ", Account Holder: " << accountHolderName
             << ", Balance: " << balance << "\n";
    }
};

int main() {
    // Array to store multiple bank accounts
    const int numAccounts = 3;
    BankAccount accounts[numAccounts] = {
        BankAccount(101, 500.0, "Stephen"),
        BankAccount(102, 1000.0, "Elisha"),
        BankAccount(103, 750.0, "Gachoka")
    };

    // Display each account's details
    cout << "Initial Account Details:\n";
    for (int i = 0; i < numAccounts; i++) {
        accounts[i].displayBalance();
    }

    // Perform deposit and withdrawal on each account
    cout << "\nPerforming transactions...\n";
    for (int i = 0; i < numAccounts; i++) {
        double depositAmount = 200.0;  // Example deposit amount
        double withdrawalAmount = 150.0;  // Example withdrawal amount

        accounts[i].deposit(depositAmount);
        accounts[i].withdraw(withdrawalAmount);
```

```
    }

    // Display each account's details after transactions
    cout << "\nAccount Details After Transactions:\n";
    for (int i = 0; i < numAccounts; i++) {
        accounts[i].displayBalance();
    }

    return 0;
}

```