# TASK 5:

**HEADER FILE:**

#pragma once

#include<iostream>

using namespace std;

class Account

{

private:

double balance;

double interestRate;

double interest;

int transactions;

public:

static int count;

Account();

Account(double, double);

void setinterestRate(double);

void makeDeposit(double);

void withdraw(double);

void calcInterest();

void incCount();

int getCount();

double getInterestRate();

double getBalance();

double getInterest();

int getTransactions();

};

**FUNCTION FILE:**

#include<iostream>

#include"Header.h"

using namespace std;

//Default Constructor

Account::Account()

{

interest = 0.0;

balance = 0.0;

transactions = 0.0;

interestRate = 4.5;

}

//Parameterized Constructor

Account::Account(double bal, double intRate)

{

this->balance = bal;

this->interestRate = intRate;

}

//Setter

void Account::setinterestRate(double intRate)

{

this->interestRate = intRate;

}

//Deposit

void Account::makeDeposit(double amount)

{

balance += amount;

}

//Withdraw

void Account::withdraw(double amount)

{

if (amount > balance)

{

cout << "ERROR! Withdrawal amount too large.";

}

else

{

balance -= amount;

transactions++;

cout << "Withdrawal successfull.";

}

}

//Interset

void Account::calcInterest()

{

interest = (balance \* interestRate) / 100;

balance += interest;

cout << "Interest added.";

}

//Count

void Account::incCount()

{

count++;

}

//Count Getter

int Account::getCount()

{

return count;

}

//Interest Rate Getter

double Account::getInterestRate()

{

return interestRate;

}

//Balance Getter

double Account::getBalance()

{

return balance;

}

//Interest Getter

double Account::getInterest()

{

return interest;

}

//Transactions Getter

int Account::getTransactions()

{

return transactions;

}

//Static Count

int Account::count = 0;

**MAIN FILE:**#include<iostream>

#include<string>

#include "Header.h"

using namespace std;

int main()

{

Account a;

while (1)

{

int choice;

cout << "Enter your choice: " << endl;

cout << "1) Display the Account balance." << endl;

cout << "2) Display the number of transactions." << endl;

cout << "3) Display interest earned for this period." << endl;

cout << "4) Make a deposit." << endl;

cout << "5) Make a withdrawal." << endl;

cout << "6) Add interest for this period." << endl;

cout << "7) Exit the program." << endl << endl;

cout << "Number of times program has taken choice: " << a.getCount() << endl;

cin >> choice;

if (choice == 1)

{

Account::count++;

cout << "The current balace is $" << a.getBalance() << endl;

cout << endl;

}

else if (choice == 2)

{

Account::count++;

cout << "There have been " << a.getTransactions() << " transactions." << endl;

cout << endl;

}

else if (choice == 3)

{

Account::count++;

cout << "Interest earned for this period: $" << a.getInterest() << endl;

cout << endl;

}

else if (choice == 4)

{

Account::count++;

double amount;

cout << "Enter the amount of the deposit: ";

cin >> amount;

a.makeDeposit(amount);

cout << endl;

}

else if (choice == 5)

{

Account::count++;

double amount;

cout << "Enter the amount of the withdrawal: ";

cin >> amount;

a.withdraw(amount);

cout << endl;

}

else if (choice == 6)

{

Account::count++;

a.calcInterest();

cout << endl;

}

else if (choice ==7)

{

Account::count++;

cout << "Program exited successfully.";

break;

}

else

{

Account::count++;

cout << "Wrong choice! Please enter again!";

cout << endl << endl;

}

}

}

# OUTPUT:

