Ulises Ramirez

[Ulises.ramirez@snhu.edu](mailto:Ulises.ramirez@snhu.edu)

Southern New Hampshire University

***Airgead Banking pseudocode***

include H file named calculateBalance

create function named noMonthlyPayment with years, Interest, amount passed to it.

Create a double variable named totalAmount and assign it amount.

Cout the display banner

Start of For loop until its less than years

Assign Amount to totalAmount times interest divided by 100.

Assign totalAmount by totalAmount plus amount.

Cout the year, total amount and amount

End for loop

Cout endl to give space.

End of noMonthlyPayment function

create function named MonthlyPayment with years, Interest, amount, monthlyDeposit passed to it.

Create a double variable named totalAmount and assign it amount.

Cout the display banner

Start of first For loop until its less than years

Create double named yearlyInterest and assign it 0.0.

Start of second loop until its less than 12

Assign Amount to totalAmount plus monthlyDeposit times interest divided by 100. Divided by 12.

Assign yearlyInterest by yearlyInterest plus amount.

Assign totalAmount by totalAmount plus monthlyDeposit plus amount.

End of second loop

Cout the year, total amount and amount

End first for loop

Cout endl to give space.

End of MonthlyPayment function

Start of main Function

Create from calculateBalance class called account1.

Create interest,monthly,startammt, yearnum variables

Cout the display banner

Pause

Endl for extra space

Cout the display banner

Get startAmount, monthlyDeposits. interest, and years from the user

Pause

Endl for extra space

Assign startAmount, monthlyDeposits. interest, and years to the variable account1 of class calculateBalance

Call function noMonthlyPayment and send years yearlyInterest StartAmount variables.

Call function MonthlyPayment and send years yearlyInterest StartAmount monthlydeposit variables.

Ends main cpp page.

Start of calculateBalance H file

Start of calculateBalance class

Create variables startAmount, monthlyDeposit, yearlyInterest, years in private.

Start of public

Call/create Setyears mutator and send years variable.

Call/create SetStartAmountmutator and send startamount variable.

Call/create SetmonthlyDeposit mutator and send monthlyDeposit variable.

Call/create SetyearlyInterestmutator and send yearlyInterestvariable.

Call/create GetStartAmount accessors.

Call/create GetmonthlyDeposit accessors.

Call/create GetyearlyInterest accessors.

Call/create Getyears accessors.

End of calculateBalance H file

Start of calculateBalance cpp file

Create function Setyears from class calculateBalance and assign year to years.

Create function SetStartAmount from class calculateBalance and assign StartAmounts to StartAmount.

Create function SetmonthlyDeposit from class calculateBalance and assign monthlyDeposits to monthlyDeposit.

Create function SetyearlyInterest from class calculateBalance and assign yearlyInterests to yearlyInterest.

Create function GetStartAmount from class calculateBalance and return startAmount.

Create function GetmonthlyDepositfrom class calculateBalance and return monthlyDeposit.

Create function GetyearlyInterestfrom class calculateBalance and return yearlyInterest.

Create function Getyearsfrom class calculateBalance and return years.

end of calculateBalance cpp file

end of program