//PROBLEM2\_FUNCTIONS.cpp

//Andrew Ribeiro

//October 6, 2008

//Used in: PROBLEM2.cpp

/\*File Discription & Contents

>This file containes the functions used in the file mortgage\_main.cpp.

#Contents/Functions#

################ FUNCTIONS: TO BE CALLED IN MAIN PROGRAM ######################

1.) double safeInput(string errorMessage,string prompt,double minValue)

This function allows for error checking and a customised invalid value message and value

request message. Also, this function makes shure that the users input is not less then the

minValue. Finaly, this function returns the users input when the input is valid.

2.)void highIntrestRateLongPaymentsWarning(bool &printMortgagePaymentSchedule,\

double principal,double &monthlyPayment,double &monthlyIntrest)

This function basically check the logic of the input. If the interest is large,the principal

is large, and the monthly payment is small... chanses are that the program will fall into

an infinite loop. Therefore, this function was needed to check if the mortgage could be payed

and if it could be payed but had a relatively large payment scedule(each itteration only

reducing the debt by .5 to be percise)the user was given a warning to prevent against a

long unexpected wait-perhaps the user entered in a wrong value.This function also allows the

user to change interest andmonthly payment values, or allows the user to exit the program.

3.)void sub\_calculateAndDisplayMortgagePaymentSchedule(double principal,double monthlyPayment,double monthlyIntrest)

This function prints out the mortgage payment schedule, tells the user how many months or years

it will take to pay off the mortgage, tells the user how much interest he/she would pay, and tells

the user how much the total mortgage costs- including interest.

################ HELPER FUNCTIONS: NOT TO BE CALLED IN MAIN PROGRAM ######################

4.)char safeInput(string errorMessage,string prompt,char validChar1,char validChar2)

This function is a char version of the safeInput above- the function is overloaded.

The diference between the two functions aside from the return type, is that this

function only accepts two charicters (designed for yes or no type questions). This

function also returns a valid user input.

End File Discription & Contents\*/

#include <iostream>

#include <iomanip>

#include <string>

#include <cmath>

using namespace std;

double safeInput(string errorMessage,string prompt,double minValue)

{

double userIn = 0;

cin>>userIn;

//Loop if there is errors. Be aware of the side effects.

while(!cin.good() || userIn < minValue )

{

cout<<errorMessage<<endl

<<prompt;

cin.clear();

cin.ignore(80,'\n');

cin>>userIn;

}

return userIn;

}

char safeInput(string errorMessage,string prompt,char validChar1,char validChar2)

{

//Will convert the user input to upper case charicters.

char userIn;

cin>>userIn;

userIn = toupper(userIn);

//Loop if there is errors. Be aware of the side effects.

while(!cin.good() || userIn != validChar1 && userIn != validChar2)

{

cout<<errorMessage<<endl

<<prompt;

cin.clear();

cin.ignore(80,'\n');

cin>>userIn;

}

return userIn;

}

void sub\_calculateAndDisplayMortgagePaymentSchedule(double principal,double monthlyPayment,double monthlyIntrest)

{

double remainingAmount,totalPaymentAmountWithInterest = 0;

int counterMonth = 0;

remainingAmount = principal; //At the begining of the loop the remaining amount is == to the principal.

//Print header

cout<<endl<<setw(10)<<"Month"<<setw(15)<<"Repayment"<<setw(15)<<"Remaining"<<setw(15)<<endl

<<" -----------------------------------"<<endl;

//While there is still some of the mortgage left to be payed.

while(remainingAmount > 0)

{

/\*If the remainingAmount is less than the monthly payment, pay whats left.

Else, pay the regular monthly payment\*/

counterMonth++;

if (remainingAmount < monthlyPayment)

{

//Last month

remainingAmount = (remainingAmount +(remainingAmount \* (monthlyIntrest/100)));

cout<<setw(10)<<counterMonth<<setw(15)<<remainingAmount<<setw(15)<<"0.00"<<endl;

totalPaymentAmountWithInterest += remainingAmount;

remainingAmount = 0;

}

else

{

totalPaymentAmountWithInterest += monthlyPayment;//Summing up each months payment.

remainingAmount = (remainingAmount +(remainingAmount \* (monthlyIntrest/100))) - monthlyPayment;

cout<<setw(10)<<counterMonth<<setw(15)<<monthlyPayment<<setw(15)<<remainingAmount<<endl;

}

}

if(counterMonth < 12)

{

//If it will take less than 12 months to pay off the loan.

if(counterMonth == 1)

{

//If it will take only one month, do not use "months" in the printout.

cout<<endl<<endl<<"At this rate it will take you "

<<counterMonth % 12<<" month to pay of the mortgage.\n"<<endl;

}

else

{

cout<<endl<<endl<<"At this rate it will take you "

<<counterMonth % 12<<" months to pay of the mortgage.\n"<<endl;

}

}

else

{

if(counterMonth == 12)

{

//If there is one year.

cout<<endl<<endl<<"At this rate it will take you "<<counterMonth / 12

<<" year to pay of the mortgage.\n"<<endl;

}

else if (counterMonth % 12 == 0)

{

//If there are only years, no remaining months.

cout<<endl<<endl<<"At this rate it will take you "<<counterMonth / 12

<<" years to pay of the mortgage.\n"<<endl;

}

else

{

//If there are months and years.

cout<<endl<<endl<<"At this rate it will take you "<<counterMonth / 12

<<" years and "<<counterMonth % 12<<" months to pay of the mortgage.\n"<<endl;

}

}

cout<<"You will pay a total of: "<<totalPaymentAmountWithInterest<<endl;

cout<<"Total interest payed: "<<totalPaymentAmountWithInterest - principal<<endl<<endl;

}

void highIntrestRateLongPaymentsWarning(bool &printMortgagePaymentSchedule,\

double principal,double &monthlyPayment,double &monthlyIntrest)

{

while(true)

{

if ((principal - monthlyPayment) + principal \* ( monthlyIntrest/100 ) >= principal)

{

/\* If the first months payment is subtracted from the inital principal and the intrest is added, and

the total is larger than the inital princial; the user will never be able to pay off the loan because

the principal will keep growing. \*/

int userChoice;

cout<<"\n WARNING: This loan will never be repaid. The intrest rate is too high.\n"<<endl

<<"Woud you like to adjust your interest rate(1),monthly payment(2),or 3 for both.\n"

<<"Press 1,2,3,or a number greater than 3 to exit: ";

//Get users choice

userChoice = safeInput("Invalid input.","Please enter 1,2,or any other number greater than 2: ",1);

//Allow the user to adjust the interest rate, monthly payment, or exit.

if (userChoice == 1)

{

cout<<"Enter in a lower interest rate: ";

monthlyIntrest = safeInput("Invalid interest rate!",\

"Please enter a valid interest rate(press -1 to exit): ",-1);

}

else if(userChoice == 2)

{

cout<<"Enter in a higher monthly payment: ";

monthlyPayment = safeInput("Invalid monthly payment!",\

"Please enter a valid monthly payment:(press -1 to exit): ",1);

}

else if(userChoice == 3)

{

cout<<"Enter in a lower interest rate: ";

monthlyIntrest = safeInput("Invalid interest rate!",\

"Please enter a valid interest rate(press -1 to exit): ",-1);

cout<<"Enter in a higher monthly payment: ";

monthlyPayment = safeInput("Invalid monthly payment!",\

"Please enter a valid monthly payment:(press -1 to exit): ",1);

}

else

{

//user wants to exit and not print calendar

printMortgagePaymentSchedule = false;

break;

}

}

else if( principal - (principal - monthlyPayment) + principal \* ( monthlyIntrest/100 ) <= (principal \*(0.5/100)) )

{

/\*Calculate the persentage of the principal the first payment takes off the principal. And find

how many months it will take to pay the user. This will allow the user to avoid printing out long

mortgage schedules unexpectedly.

Note: it will not be exact because the loss of persison\*/

double howManyMonthsItWillTakeToPayMortgage;

char continueToPrint;

howManyMonthsItWillTakeToPayMortgage =\

(1.0 /((principal -((principal - monthlyPayment) + principal \* ( monthlyIntrest/100 )))/principal) );

howManyMonthsItWillTakeToPayMortgage = ceil(howManyMonthsItWillTakeToPayMortgage);

/\*If the diference between the principal and the first month payment subtraction from the principal is less than 5 percent of the principal warn and prompt the user that the printout may be large, and ask them if it should contiue.\*/

cout<<"\n WARNING: the printout of the mortgage payment schedule will be about "

<<howManyMonthsItWillTakeToPayMortgage

<<endl<<" rows/months long.\n"<<endl

<<"Would you like to continue and print out the mortgage payment schedule(Y/N): ";

continueToPrint = safeInput("Invalid choice.","Please enter(Y/N): ",'Y','N');

if (continueToPrint == 'N')

{

printMortgagePaymentSchedule = false;

}

break;

}

else

{

break;

}

}

}