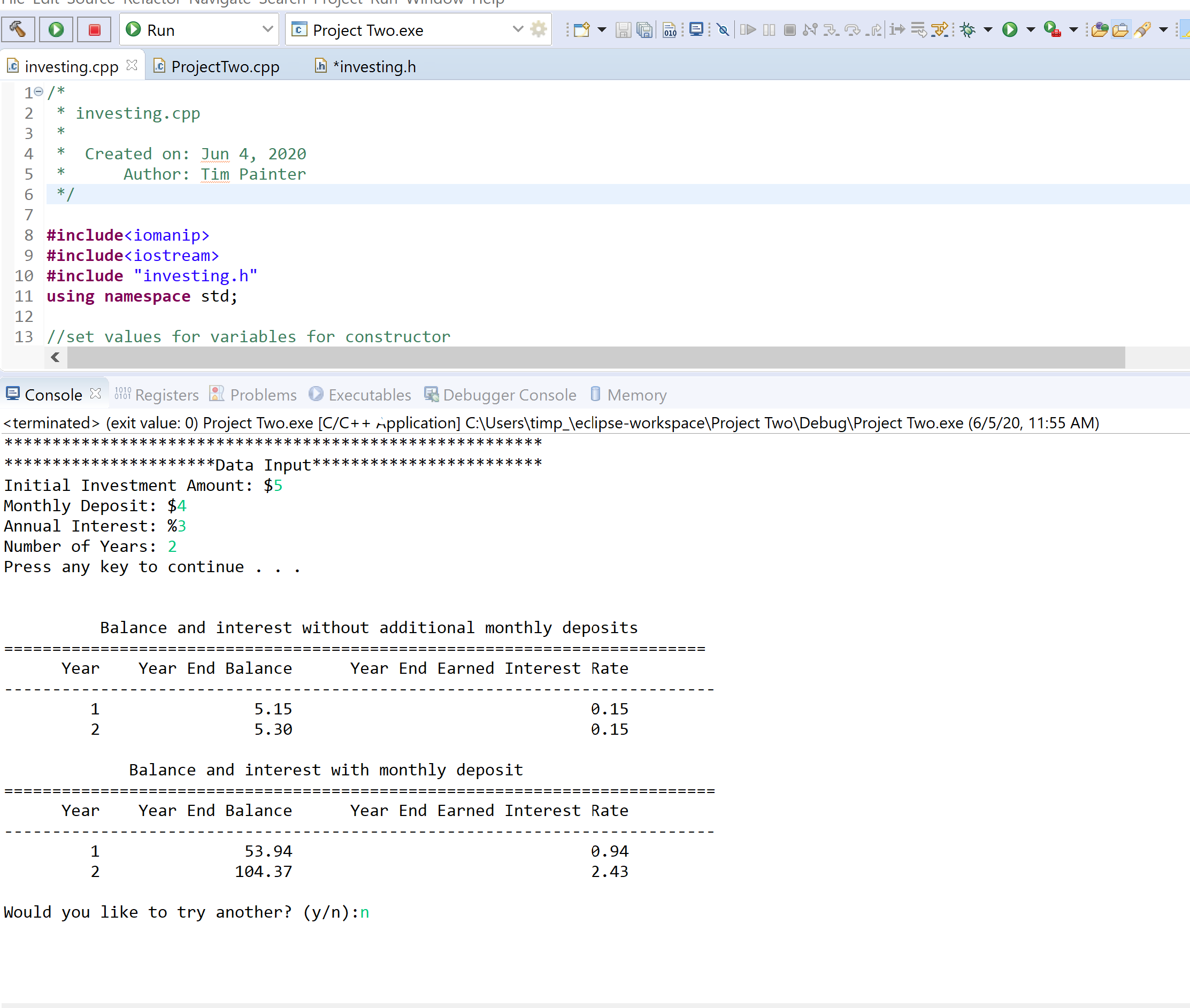
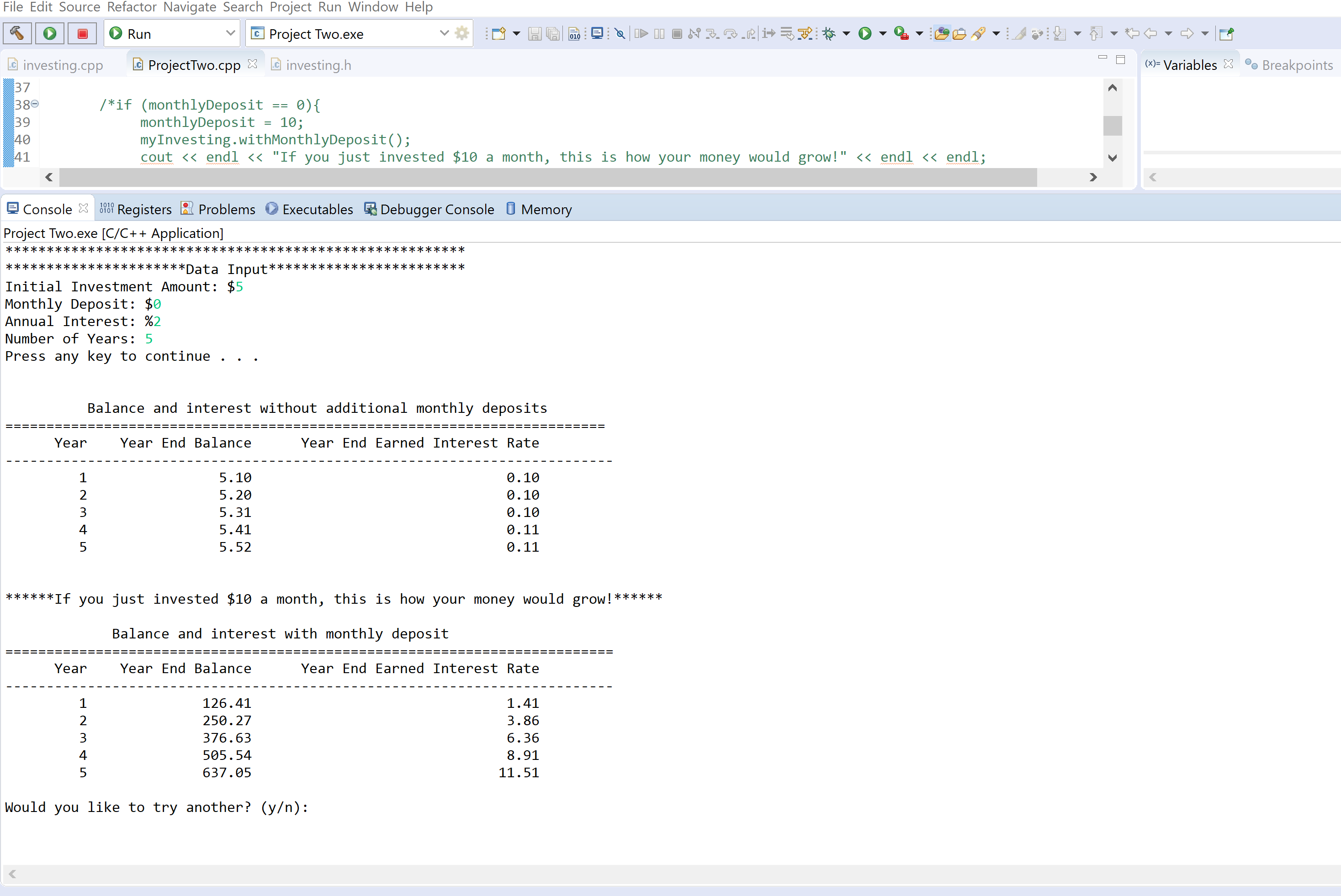
With a monthly deposit:



Without a monthly deposit (defaults to a $10 deposit to compare and encourage saving):



***CODE:***

***ProjectTwo.cpp***

/\*

\* ProjectTwo.cpp

\*

\* Created on: Jun 4, 2020

\* Author: Tim Painter

\*/

**#include** <iostream>

**#include** "investing.h"

**using** **namespace** std;

//getting user input. Displays required headers

**int** **main**(){

**while** (1){

//getting user input

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << **endl**;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Data Input\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << **endl**;

cout << "Initial Investment Amount: $";

**double** initDeposit, monthlyDeposit, interestRate;

**int** years;

cin >> initDeposit;

cout << "Monthly Deposit: $";

cin >> monthlyDeposit;

cout << "Annual Interest: %";

cin >> interestRate;

cout << "Number of Years: ";

cin >> years;

**system**("pause");//pauses system and instructs user to continue by hitting any key

//investing objects dependent on user inputs

investing myInvesting = investing(initDeposit, monthlyDeposit, interestRate, years);

cout << **endl**;

myInvesting.noMonthlyDeposit();

cout << **endl**;

**if**(monthlyDeposit > 0){

myInvesting.withMonthlyDeposit();

}

**else** **if** (monthlyDeposit == 0){

cout << **endl** << "\*\*\*\*\*\*If you just invested $10 a month, this is how your money would grow!\*\*\*\*\*\*" << **endl** << **endl**;

monthlyDeposit = 10;

investing myInvesting = investing(initDeposit, monthlyDeposit, interestRate, years);

myInvesting.withMonthlyDeposit();

}

//give user option to do another or exit

cout << **endl** << "Would you like to try another? (y/n):";

string choice;

cin >> choice;

**if**(choice != "y"){

**break**;

}

cout << **endl**;

}

**return** 0;

}

***Investing.cpp***

/\*

\* investing.cpp

\*

\* Created on: Jun 4, 2020

\* Author: Tim Painter

\*/

**#include**<iomanip>

**#include**<iostream>

**#include** "investing.h"

**using** **namespace** std;

//set values for variables for constructor

**investing::investing**(**double** initInvestment, **double** monthlyInvestment, **double** rate, **int** years){

**this**->initialInvestment = initInvestment;

**this**->monthlyInvestment = monthlyInvestment;

**this**->interestRate = rate;

**this**->numYears = years;

}

//destructor

**investing::~investing**(){

}

//prints header and report for no monthly deposit

**void** **investing::noMonthlyDeposit**(){

cout << " Balance and interest without additional monthly deposits" << **endl**;

cout << "=========================================================================" << **endl**;

cout << setw(10) << "Year" << setw(20) << "Year End Balance" << setw(35) << "Year End Earned Interest Rate" << **endl**;

cout << "--------------------------------------------------------------------------" << **endl**;

**int** currentYear = 1;

**double** yearEndBalance = **this**->initialInvestment;

//calculates interest monthly and compounded

**while**(currentYear <= **this**->numYears){

**double** intEarned = yearEndBalance \* **this** -> interestRate / 100;

yearEndBalance += intEarned;

cout << right << setw(10) << currentYear << fixed << setprecision(2) << setw(20) << yearEndBalance << setw(35) << intEarned << **endl**;

currentYear++;

}

}

//prints header and report for monthly deposit

**void** **investing::withMonthlyDeposit**(){

cout << " Balance and interest with monthly deposit" << **endl**;

cout << "==========================================================================" << **endl**;

cout << setw(10) << "Year" << setw(20) << "Year End Balance" << setw(35) << "Year End Earned Interest Rate" << **endl**;

cout << "--------------------------------------------------------------------------" << **endl**;

**int** currentYear = 1;

**double** yearEndBalance = **this**->initialInvestment;

//calculates interest monthly and compounded

**while**(currentYear <= **this**->numYears){

**int** month = 1;

**double** intEarned = 0.0;

**double** monthEndBalance = yearEndBalance;

**while**(month <= 12){

monthEndBalance += **this**->monthlyInvestment;

**double** monthlyInterest = monthEndBalance \* **this**-> interestRate/(100\*12);

intEarned += monthlyInterest;

monthEndBalance += monthlyInterest;

month++;

}

yearEndBalance = monthEndBalance;

cout << right << setw(10) << currentYear << fixed << setprecision(2) << setw(20) << yearEndBalance << setw(35) << intEarned << **endl**;

currentYear++;

}

}

***Investing.h***

**#ifndef** INVESTING\_H\_

**#define** INVESTING\_H\_

//create class and assign variables

**class** investing{

**public**:

**investing**(**double** intialInvestment, **double** monthlyInvestment, **double** rate, **int** years);

**virtual~investing**();

**void** **noMonthlyDeposit**();

**void** **withMonthlyDeposit**();

**private**:

**double** initialInvestment;

**double** monthlyInvestment;

**int** numYears;

**double** interestRate;

};

**#endif** /\* INVESTING\_H\_ \*/