//course comsc-200

//status : completed

// lab3a

//

// Created by Jeff on 8/29/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

#include <iostream>

#include <sstream> // stringstream

#include <string>

using namespace std;

void menu(){

cout<< "------------Fool\_Proof\_Meanu------------" << endl;

cout<< "1-Enter a line of text" << endl

<<"2-Enter a text phrse" <<endl

<<"3-Enter a integer number "<<endl

<<"4-Enter a float number " <<endl

<<"q-Quite"<<endl

<<"enter your choice: ";

}

int main()

{

int i;

string input,choice,text;

double d;

bool stay = true;

while(stay) {

menu();

cout << endl;

cin >> choice;

cin.ignore(1000,10);

if(choice.size()==1){

char ch = choice[0];

switch(ch){

case '1':{

cout<< endl << " enter a line of text :" <<endl;

while(true){

getline(cin, input);

stringstream ss(input);

text = ss.str();

if (!ss.fail())break; // not an integer

cout << " => Error in converting the string !\n"<<"retry: "<< endl;

}

cout << "you entered a line of text : " << endl<< text << endl;

break ;

}

case '2':

{

cout<< " enter a word of text :"<<endl;

while(true){

getline(cin, input);

//cin.ignore(1000,10);

stringstream ss(input);

ss >> text; // string type

if (!ss.fail())break;// not an strying type

cout << " error in converting the string you entered \n" << endl <<"try again : " <<endl;

}

cout << "you have entered : "<< endl <<text << endl;

break;

}

case '3':{

cout << "\nEnter a intger number: ";

while(true){

getline(cin, input);

stringstream ss(input);

ss >> i;

if (!ss.fail()) break; // not an integer

cout << " => Can not find a intger number!\n"<< endl << "try it again :"<<endl;

}

cout << " => Found this intger number: " << i << endl;

break;

}

case '4':

{

cout << "\nEnter a floating number: ";

while(true){

getline(cin, input);

cout << " You have entered: " << input << endl;

stringstream ss(input);

//num = ss.float();

ss >> d;

if (!ss.fail()) break;// not an integer

cout << " => Can not find a floating number!\n";

}

cout << " => Found this floating number: " << d << endl;

break;

}

case 'q' :

case 'Q' :

stay = false;

break ;

default:

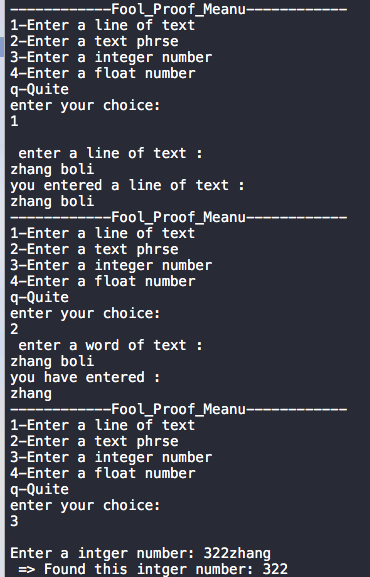
cout << "command are not supported " <<endl;

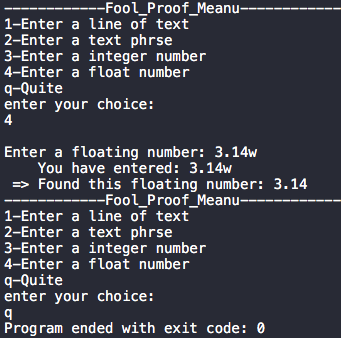
}

}

}

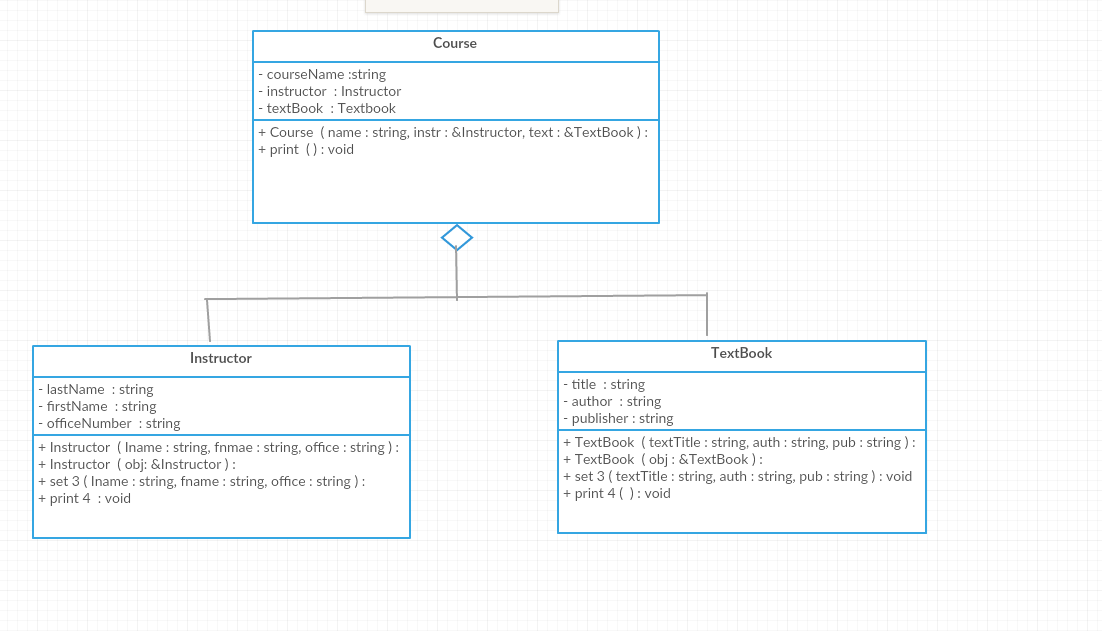
}





//lab3b

//completed



//lab3c

//completed

//

// main.cpp

// lab3c

//

// Created by Jeff on 8/31/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

// Chapter 14, Programming Challenge 1: Number Class

#include <iostream>

#include <string>

using namespace std;

// Declaration of Numbers class

class Numbers

{

private:

int number; // To hold a number

// Static arrays to hold words

static string lessThan20[20];

static string tens[10];

static string hundred;

static string thousand;

public:

// Constructor

Numbers(int x){ number = x;}

// Function to print the words for the number

void print();

};

// Static member variables must be defined

// outside of the class

string Numbers::lessThan20[20] =

{ "zero", "one", "two", "three", "four", "five",

"six", "seven", "eight", "nine", "ten",

"eleven", "twelve", "thirteen", "fourteen",

"fifteen", "sixteen", "seventeen", "eighteen",

"nineteen",

};

string Numbers::tens[10] =

{ "zero", "ten", "twenty", "thirty", "forty",

"fifty", "sixty", "seventy", "eighty", "ninety",

};

string Numbers::hundred = "hundred";

string Numbers::thousand = "thousand";

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// The print fucntion prints the English words \*

// for the number \*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Numbers::print()

{

// Residue holds what remains to be printed.

int residue = number;

// Take care of thousands, if any.

int n\_thousands = residue/1000;

residue = residue % 1000;

if (n\_thousands > 0)

{

cout << " " << lessThan20[n\_thousands];

cout << " thousand ";

}

// Fill the blank

// Take care of hundreds, if any.

int n\_hundreds = residue/100;

residue = residue % 100;

if (n\_hundreds > 0)

{

cout << " " << lessThan20[n\_hundreds];

cout << " hundreds ";

}

// Take care of anything less than 20

int n\_tens = residue/10;

residue = residue % 10;

if (n\_tens > 1)

{

cout << " " << tens[n\_tens];

cout << " " << lessThan20[residue];

}

else if (n\_tens == 1)

cout << " " << lessThan20[residue + n\_tens \* 10];

// else if

}

// Demo program

int main()

{

int number;

// Tell user what the program does.

cout << "Translates whole dollar amounts into words for"

<< "the purpose of writing checks.\n"

<< "Entering a negative number terminates the program.\n"

<< "Enter an amount (less than 20000)for be translated into words: ";

cin >> number;

while (number >= 0)

{

// Create a Numbers object.

Numbers n(number);

// Print the English description.

n.print();

// Get another number.

cout << "\nEnter another number: ";

cin >> number;

}

return 0;

}

//lab3c

//completed

//

// main.cpp

// lab3c

//

// Created by Jeff on 8/31/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

// Chapter 14, Programming Challenge 1: Number Class

#include <iostream>

#include <string>

using namespace std;

// Declaration of Numbers class

class Numbers

{

private:

int number; // To hold a number

// Static arrays to hold words

static string lessThan20[20];

static string tens[10];

static string hundred;

static string thousand;

public:

// Constructor

Numbers(int x){ number = x;}

// Function to print the words for the number

void print();

};

// Static member variables must be defined

// outside of the class

string Numbers::lessThan20[20] =

{ "zero", "one", "two", "three", "four", "five",

"six", "seven", "eight", "nine", "ten",

"eleven", "twelve", "thirteen", "fourteen",

"fifteen", "sixteen", "seventeen", "eighteen",

"nineteen",

};

string Numbers::tens[10] =

{ "zero", "ten", "twenty", "thirty", "forty",

"fifty", "sixty", "seventy", "eighty", "ninety",

};

string Numbers::hundred = "hundred";

string Numbers::thousand = "thousand";

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// The print fucntion prints the English words \*

// for the number \*

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Numbers::print()

{

// Residue holds what remains to be printed.

int residue = number;

// Take care of thousands, if any.

int n\_thousands = residue/1000;

residue = residue % 1000;

if (n\_thousands > 0)

{

cout << " " << lessThan20[n\_thousands];

cout << " thousand ";

}

// Fill the blank

// Take care of hundreds, if any.

int n\_hundreds = residue/100;

residue = residue % 100;

if (n\_hundreds > 0)

{

cout << " " << lessThan20[n\_hundreds];

cout << " hundreds ";

}

// Take care of anything less than 20

int n\_tens = residue/10;

residue = residue % 10;

if (n\_tens > 1)

{

cout << " " << tens[n\_tens];

cout << " " << lessThan20[residue];

}

else if (n\_tens == 1)

cout << " " << lessThan20[residue + n\_tens \* 10];

// else if

}

// Demo program

int main()

{

int number;

// Tell user what the program does.

cout << "Translates whole dollar amounts into words for"

<< "the purpose of writing checks.\n"

<< "Entering a negative number terminates the program.\n"

<< "Enter an amount (less than 20000)for be translated into words: ";

cin >> number;

do

{

// Create a Numbers object.

Numbers n(number);

// Print the English description.

n.print();

// Get another number.

cout << "\nEnter another number: ";

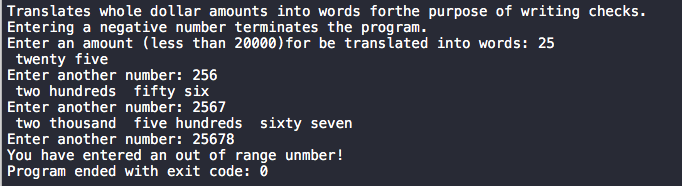
cin >> number;

}while(number <20000);

cout<< "You have entered an out of range unmber! " <<endl;

return 0;

}



//lab3d

//completed

//somehow it’s working on Xcode, but It works on terminal.

//

// SavingsAcct.cpp

// lab 3d

//

// Created by Jeff on 8/31/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

#include <stdio.h>

#inlcude "SavingsAcct.h"

//#import "SavingsAcct.h"

double SavingsAcct::annualInterestRate = 0.0;

void SavingsAcct::calculateMonthlyInterest(){

balance += balance\*annualInterestRate /12.0;

}

void SavingsAcct::modifyInterestRate(double r){

if(0.0 <= r && r <= 1.0)

annualInterestRate = r ;

}

void SavingsAcct::printBalance(){

std::cout << std::fixed << std::setprecision(2) << balance ;

}

//

// SavingsAcct.h

// lab 3d

//

// Created by Jeff on 8/31/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

#ifndef SAVINGSACCT\_H

#define SAVINGSACCT\_H

class SavingsAcct{

private:

static double annualInterestRate;

double balance;

public:

SavingsAcct(double b)

{

balance = b;

};

void calculateMonthlyInterest();

static void modifyInterestRate(double);

void printBalance();

};

/\*

double SavingsAcct::annualInterestRate = 0.0;

void SavingsAcct::calculateMonthlyInterest(){

balance += balance\*annualInterestRate /12.0;

}

void SavingsAcct::modifyInterestRate(double r){

if(0.0 <= r && r <= 1.0)

annualInterestRate = r ;

}

void SavingsAcct::printBalance(){

std::cout << std::fixed << std::setprecision(2) << balance ;

}

\*/

#include "SavingsAcct.cpp"

#endif /\* SavingsAcct\_h \*/

//

// main.cpp

// lab 3d

//

// Created by Jeff on 8/31/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

#include <iostream>

#include <iomanip>

#include "SavingsAcct.h"

using namespace std;

int main()

{

SavingsAcct saver1( 2000.0 );

SavingsAcct saver2( 3000.0 );

SavingsAcct::modifyInterestRate( .03 ); // change interest rate

cout << "Initial balances:\nSaver 1: ";

saver1.printBalance();

cout << "\tSaver 2: ";

saver2.printBalance();

saver1.calculateMonthlyInterest();

saver2.calculateMonthlyInterest();

cout << "\n\nBalances after 1 month's interest applied at .03:\n"

<< "Saver 1: ";

saver1.printBalance();

cout << "\tSaver 2: ";

saver2.printBalance();

SavingsAcct::modifyInterestRate( .04 ); // change interest rate

saver1.calculateMonthlyInterest();

saver2.calculateMonthlyInterest();

cout << "\n\nBalances after 1 month's interest applied at .04:\n"

<< "Saver 1: ";

saver1.printBalance();

cout << "\tSaver 2: ";

saver2.printBalance();

cout << endl;

} // end main

