/\*

\* File: testmain.cpp

\* Author: Tracy Quintos

\* Created on December 6, 2014

\* Purpose: project 2 17A 48130

\*/

include iostream

include cstdlib

include vector //challenge mode pay out list

include cstring //case 5 "phone a friend"

include fstream //reading for file Challenger mode rules

include "MyFunctions.h" //header that holds functions for Main Game

include "Mother1.h" //base class for challenger mode

include "Daughter1.h" //derived class for challenger mode, inherits Mother1::inputException1 question 1, and other functions

using namespace std

//User Libraries

//Global Constant

//Function Prototype

//Execution starts here!

/\*

\*

\*/

struct player //function with structure

//structure for gathering player information after when GAME OVER

char firstName[20] //holds first name

char lastName[20] //holds last name

char address[20] //address info

char addresso[20] //address info

char addressp[10] //address info

char occupation[20] //holds occupation

age //holds age

finalScore //holds final money po received

main argc, char\*\* argv

//

const arraySize = 2 //size of the array that holds how many ID s there are

tests[arraySize] = 2123629, 2152573 //array to hold ID s

accountN = 0 //holds the account numbers

results = 0 //holds what the account is in the elementinside array

"\*Welcome to 17A Project 2 Fall2014\*"

" Please enter a valid ID " //Asks user to input their ID number

accountN //variable for ID

results = searchListtests, arraySize, accountN //linear search result

if results == -1//if resultID number is -1 false msg will show

"SORRY, INVALID ENTRY, PLEASE TRY AGAIN.\n"

//if user ID is not valid this message will show, user has to try again.

else // else the user will be brought to the main menu to begin game, see rules, or quit.

mainMenu //main menu function

//Exit stage right

return 0

mainCHMenu // main challenge mode menu function

Daughter1 teri //derived daughter class 'teri'

//displays money po system in "case 5" 1 D array

//has 5 money awards, each question is worth $300k -grand total $1.5

moneyP[1][5] = 300000, 300000, 300000, 300000, 300000

\*pMoney[2]

\*pMoney = moneyP[0]

choice //holds the choices for menu

number1 //holds choice 4 Phone number of friend life line

ofstream outputFile //writes phone a friend life line phone number CASE 4

ifstream nameFile //ifstream namefile to read the RULES.txt

string input //string input, inputs the rules from txt onto console

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n" //Opening title if function main menu is opened

" \*WHO WANTS\*\n"

" \*TO BE A\*\n"

" \*MILLIONAIRE\*\n"

" \*CHALLANGE MODE\*\n"

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

"1. START GAME \n" //choice 1 allows player to start the game

"2. RULES \n" //choice 2 opens the RULES.txt file onto console, rules can be read through here

"3. QUIT CHALLANGE MODE \n" //choice 3 quits the game

"4. PRIZE MONEY LIST \n" //choice 4 display money prize

"5. PHONE-A-FRIEND CHALLENGE MODE \n"

"PLEASE ENTER CHOICE: " // msg for a menu choice

choice //inputs choice

switch choice //switch statement which holds the choice inputs

case 1: //case 1 or START GAME will automatically call question 1

teri.inputException1 //challenger mode question 1 being called as a derived class 'teri' inheriting from mother1

break

case 2:

"RULES PLACE HOLDER\n" //case 2 for rules

nameFile.open"RULES2.txt" //opens file "RULES.txt"will be placed in same destination as the cpp file

if nameFile //fstream to read from RULES.txt file

getlinenameFile, input //namefile, input reads file

while nameFile

input

getlinenameFile,input //getline gets the from the namefils it's input which holds the rules to the game

nameFile.close //file close after opening txt

else //else statement if txt cannot be read

"ERROR:cannot open file.\n" //incase the file cannot be read this message will be shown

break

case 3: //case 3 will a message that you have quit the game

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

"YOU CHOOSE TO QUIT CHALLENGE MODE, HAVE A NICE DAY.\n"

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

mainMenu

break

case 4: //case 5 will pr out the money po s "pay out structure"

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

"\*\*Challenge Mode Payout Structure\*\*"

displayCHNumbers

mainCHMenu

break

case 5:

phoneCHfriend

break

default:

"INVALID OPTION, RETURNING TO NORMAL GAME MENU \n" //in the event the user enters in an invalid key for choice

// an option to try again will be given

mainMenu

displayCHNumbers //stl vector

//displays the money po "pay out structure" for challenge mode

vector tracyVector

tracyVector.push\_back300000 //push\_back money prizes questions 1-5

tracyVector.push\_back400000

tracyVector.push\_back500000

tracyVector.push\_back600000

tracyVector.push\_back1500000

for unsigned i = 0 i tracyVector.size i++ //for loop to display the vector

"Money Po : " "$" tracyVector[i] " \n"

phoneCHfriend //phone a friend life line utilizing cstring, CHALLENGE user must memorize contact number in order to use life-line

.ignore //when case 5 in challenger mode menu is chosen, this allows for user to input

const NUM\_PHONES = 5 // 5 contacts

const LENGTH = 30 //length of each char is set to 30

char contacts[NUM\_PHONES][LENGTH] = //given information for player to use, SEE CHALLENGE MODE RULES

"TQ Tracy Qu os 9514439475",

"ML Mark Lehr 9518527417",

"KW Kelly Wingate 9517479632",

"GW George Wreckts 9513261457",

"KP Kenny Phung 9518547145"

char lookUp[LENGTH]

char \*strPtr = NULL

index

"Enter the initials of a contact: "

.getlinelookUp, LENGTH

for index = 0 index NUM\_PHONES index++

strPtr = strstrcontacts[index], lookUp

if strPtr != NULL

break

if strPtr != NULL

contacts[index]

mainCHMenu //after contact info is given main challenger menu will display

else

"No matching contact was found.\n" //msg will display if contact given is not found

initializeplayer s[] //initializing structure "player" function

// information gathering

i = 0

s[i].firstName //first name of player

s[i].lastName //last name of player

s[i].address //first part of address i.e. 123 ect.

s[i].addresso //second part of address i.e. "Maple" ect

s[i].addressp // street type i.e. court, bld, way ect

s[i].occupation //occupation of player

s[i].age //age of player

s[i].finalScore //final score of game

inputplayer s[] //function to input structure "player" information

//displays the prompts player to input asked information

i = 0

"Please enter your first space last name: \n"

s[i].firstName s[i].lastName

"Please your enter address: 125 Appleton rd. \n"

s[i].address s[i].addresso s[i].addressp

"Please enter occupation: \n"

s[i].occupation

"Please enter your age: \n"

s[i].age

"Please enter your final score: \n"

s[i].finalScore

pr player s[] //Functions with structures function pr

i = 0

" "

" THANKS FOR PLAYING! MONEY WILL BE SENT TO: \n"

"Contestant: " s[i].firstName " " s[i].lastName

"Address: " s[i].address " " s[i].addresso " " s[i].addressp

"Occupation: " s[i].occupation

"Age: " s[i].age " years old \n"

"Final Score: " s[i].finalScore

" \n"

winlosefunction //function that calls functions with structures //initialize and input functions

player info[1]

initialize info

input info

winprint //function that calls functions with structures //pr function

player info[1]

initialize info

print info