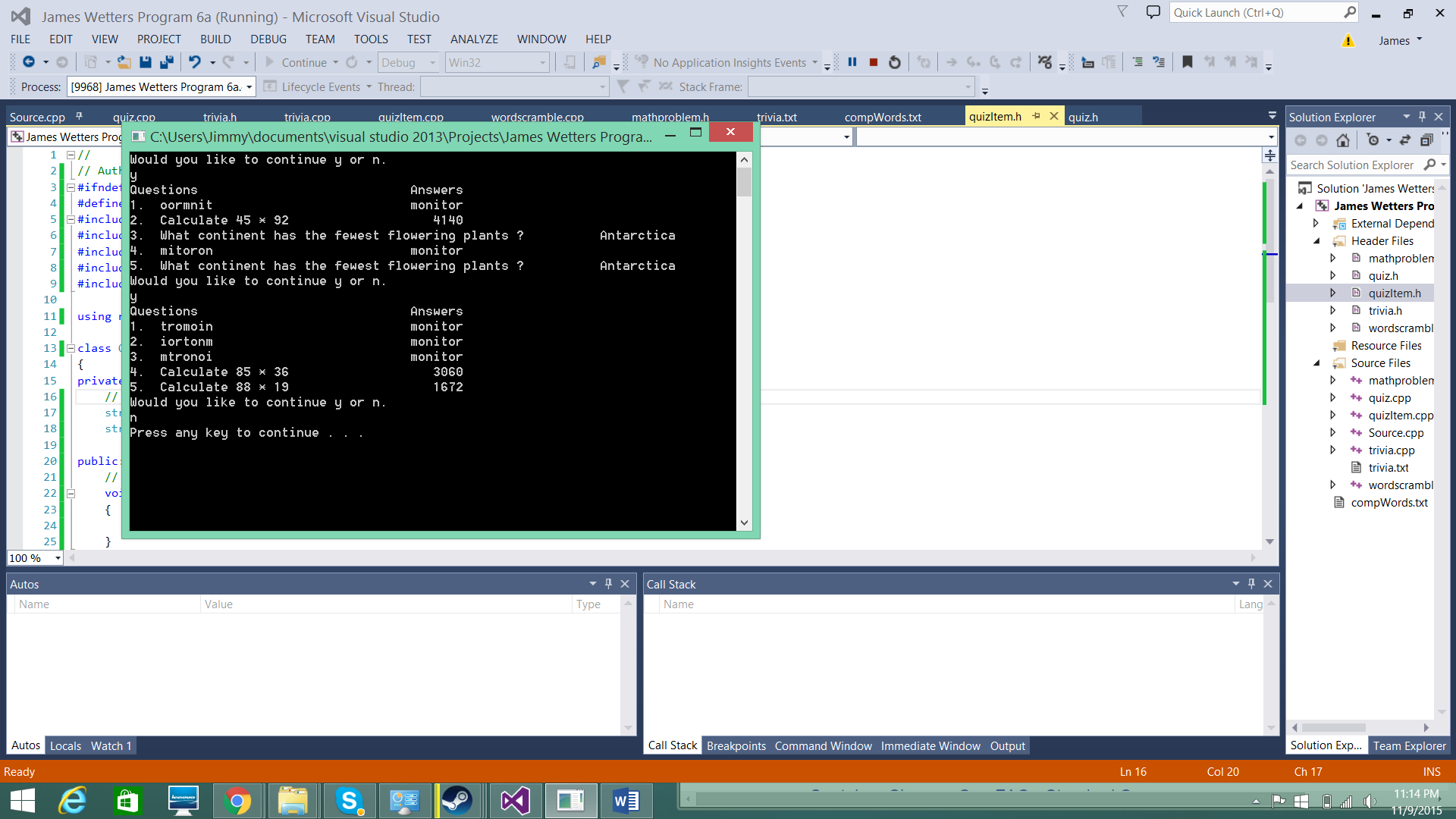
**Program 6**



Would you like to continue y or n.

y

Questions Answers

1. oormnit monitor

2. Calculate 45 \* 92 4140

3. What continent has the fewest flowering plants ? Antarctica

4. mitoron monitor

5. What continent has the fewest flowering plants ? Antarctica

Would you like to continue y or n.

y

Questions Answers

1. tromoin monitor

2. iortonm monitor

3. mtronoi monitor

4. Calculate 85 \* 36 3060

5. Calculate 88 \* 19 1672

Would you like to continue y or n.

Source

// Program 6 Main

// This program creates a quiz consiting of math, triva

// and word scramble problems

// Author James Wetters

#include <iostream>

#include <iomanip>

#include <fstream>

#include <string>

#include <cstdlib>

#include "quiz.h"

using namespace std;

// Constants

const int MAXARRAYROWS = 40;

const int MAXARRAYCOLUMS = 2;

// Prototypes

void menu();

void fileInTrivia(string theArray[MAXARRAYROWS][MAXARRAYCOLUMS], int& numElems);

void fileInCompWords(string theArray[MAXARRAYROWS], int& numElems);

int main()

{

// Get and report system clock time

time\_t seconds;

time(&seconds);

// Set random number generator seed value to system clock

srand((unsigned int)seconds);

// Menu

menu();

// Exit

system("pause");

return 0;

}

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

// Menu

//

//

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

void menu()

{

// Initilize variables

string triviaArray[MAXARRAYROWS][MAXARRAYCOLUMS];

string compWords[MAXARRAYROWS];

int triviaGoodData = 0, compWordsGoodData = 0;

char select = 'n';

// Get Arrays

fileInTrivia(triviaArray, triviaGoodData);

fileInCompWords(compWords, compWordsGoodData);

do

{

// New question HERE

cout << "Would you like to continue y or n." << endl;

cin >> select;

// If yes do a quiz

if (select == 'y' || select == 'Y')

{

// Run a quiz

Quiz a;

a.print();

}

// Check to continue

} while (select == 'y' || select == 'Y');

}

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

// Trivia File In

//

//

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

void fileInTrivia(string theArray[MAXARRAYROWS][MAXARRAYCOLUMS], int& numElems)

{

// Open File

ifstream inputFile("trivia.txt");

// Test File

if (inputFile.fail())

{

cout << "Problem opening file";

system("pause");

exit(-1);

}

// variables

string temp;

int i = 0, goodData = 0;

// Priming read

getline(inputFile, temp);

// Read in data file

while (!inputFile.eof())

{

// First Read

theArray[i][0];

// Read in secound line

getline(inputFile, temp);

theArray[i][1] = temp;

i++; // Increase goodData by 1

// Read in first line

getline(inputFile, temp);

}

// Number of good data elems

goodData = i;

// Close file

inputFile.close();

}

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

// Word Comp File In

//

//

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

void fileInCompWords(string theArray[MAXARRAYROWS], int& numElems)

{

// Open File

ifstream inputFile("compWords.txt");

// Test File

if (inputFile.fail())

{

cout << "Problem opening file";

system("pause");

exit(-1);

}

// variables

string temp;

int i = 0, goodData = 0;

// Priming read

getline(inputFile, temp);

// Read in data file

while (!inputFile.eof())

{

// First Read

theArray[i];

// Read in secound line

getline(inputFile, temp);

theArray[i] = temp;

i++; // Increase goodData by 1

// Read in first line

getline(inputFile, temp);

}

// Number of good data elems

goodData = i;

// Close file

inputFile.close();

}

// Quiz

// Author James Wetters

#ifndef QUIZ\_H

#define QUIZ\_H

#include <iostream>

#include <string>

#include <cstdlib>

#include <ctime>

#include "quizItem.h"

using namespace std;

// Initilize constants

const int PMAXARRAY = 5;

class Quiz

{

private:

// DATA MEMBERS

QuizItem \*pArray[PMAXARRAY];

public:

Quiz();

~Quiz();

// MEMBER FUNCTIONS

void print();

};

#endif

// Quiz

// Author James Wetters

#include <iostream>

#include <iomanip>

#include <string>

#include <cstdlib>

#include <ctime>

#include "quiz.h"

#include "mathproblem.h"

#include "wordscramble.h"

#include "trivia.h"

using namespace std;

// Constants

const int MAXARRAYROWS = 40;

const int MAXARRAYCOLUMS = 2;

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

// Constructor

//

// Selects 5 random questions

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

Quiz::Quiz()

{

// Initilize Variables

int select = 0;

for (int i = 0; i < PMAXARRAY; i++)

{

// Generate 1 random number 1-3

select = rand() % 3 + 1;

switch (select)

{

case 1:

// Math Problem

// Set pointer to a math problem

pArray[i] = new MathProblem;

// Generate a math problem

pArray[i]->generateQuestion();

break;

case 2:

// Word Scramble Problem

// Set pointer to a word scramble Problem

pArray[i] = new WordScrambleProblem;

// Generate a word scramble problem

pArray[i]->generateQuestion();

break;

//case 3:

default:

// Trivia Problem

// Set pointer to a triva problem

pArray[i] = new TriviaProblem;

// Generate a triva problem

pArray[i]->generateQuestion();

break;

}

}

}

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

// Print quiz//

//

// Prints all problems with pointers in the 5 question array

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

void Quiz::print()

{

// Print header

cout << "Questions" << setw(35) << "Answers" << endl;

// Print trivia questions

for (int i = 0; i < PMAXARRAY; ++i)

{

// Print problem number

cout << i + 1 << ". ";

// Pint problem

pArray[i]->print();

}

}

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

// Quiz deconstructor

//

//

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

Quiz::~Quiz()

{

// Loop through pointer array

for (int i = 0; i < PMAXARRAY; i++)

{

// Delete each pointer

delete pArray[i];

}

}

// Quiz Item

// Author James Wetters

#ifndef QUIZITEM\_H

#define QUIZITEM\_H

#include <iostream>

#include <iomanip>

#include <string>

#include <cstdlib>

#include <ctime>

using namespace std;

class QuizItem

{

private:

// DATA MEMBERS

string question;

string answer;

public:

// SETS

void setQuestion(string change)

{

question = change;

}

void setAnswer(string change)

{

answer = change;

}

// GETS

string getQuestion() const

{

return question;

}

string getAnswer() const

{

return answer;

}

// MEMBER FUNCTIONS

virtual void generateQuestion() = 0;

void print();

};

#endif

// Quiz Item

// Author James Wetters

#include "quizItem.h"

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

// Constructor

//

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

void QuizItem::print()

{

cout << setw(20) << left << question;

cout << setw(20) << right << answer << endl;

}

// Word Scramble

// Author James Wetters

#ifndef MATHPROBLEM\_H

#define MATHPROBLEM\_H

#include <iostream>

#include <string>

#include <cstdlib>

#include <ctime>

#include "quizItem.h"

using namespace std;

class MathProblem : public QuizItem

{

private:

// DATA MEMBERS

int firstNum, secoundNum;

public:

// SETS

void setFirstNum(int change)

{

firstNum = change;

}

void setSecondNum(int change)

{

secoundNum = change;

}

// GETS

int getFirstNum() const

{

return firstNum;

}

int getSecondNum() const

{

return secoundNum;

}

// MEMBER FUNCTIONS

MathProblem();

void generateQuestion();

void addition();

void multiplication();

};

#endif

// Math Problem

// Author James Wetters

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "mathproblem.h"

using namespace std;

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

// Constructor

//

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

MathProblem::MathProblem()

{

}

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

// Virtual Generate Question Math Problem

//

// Randomly selects an addition or multiplication problem

// Sends back problem in the form of a question and answer

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

void MathProblem::generateQuestion()

{

int select;

// Generate 1 random number 1-3

select = rand() % 2;

// If select == 1 then do addition problem otherwise do multiplication problem

if (select == 1)

{

addition();

}

else

{

multiplication();

}

}

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

// Addition

//

// Creates an addition problem

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

void MathProblem::addition()

{

// Initilize Variables

string temp1, temp2, tempQuestion, tempAnswer;

int answer;

// Generate 1 random number 100-999

setFirstNum( rand() % 899 + 100);

// Generate 1 random number 100-999

setSecondNum(rand() % 899 + 100);

// Question

char numstr1[10], numstr2[10];

\_itoa(getFirstNum(), numstr1, 10);

\_itoa(getSecondNum(), numstr2, 10);

temp1 = numstr1;

temp2 = numstr2;

// Create question

tempQuestion = "Calculate " + temp1 + " + " + temp2;

// Set question

setQuestion(tempQuestion);

// Calculate answer

answer = getFirstNum() + getSecondNum();

// Set answer to temp string

\_itoa(answer, numstr1, 10);

tempAnswer = numstr1;

// Set answer

setAnswer(tempAnswer);

}

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

// Multiplication

//

// Creates a multiplication problem

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

void MathProblem::multiplication()

{

// Initilize Variables

string temp1, temp2, tempQuestion, tempAnswer;

int answer;

// Generate 1 random number 100-999

setFirstNum(rand() % 89 + 10);

// Generate 1 random number 100-999

setSecondNum(rand() % 89 + 10);

// Question

char numstr1[10], numstr2[10];

\_itoa(getFirstNum(), numstr1, 10);

\_itoa(getSecondNum(), numstr2, 10);

temp1 = numstr1;

temp2 = numstr2;

// Create question

tempQuestion = "Calculate " + temp1 + " \* " + temp2;

// Set question

setQuestion(tempQuestion);

// Calculate answer

answer = getFirstNum() \* getSecondNum();

// Set answer to temp string

\_itoa(answer, numstr1, 10);

tempAnswer = numstr1;

// Set answer

setAnswer(tempAnswer);

}

// Word Scramble Problem

// Author James Wetters

#ifndef WORDSCRAMBLEPROBLEM\_H

#define WORDSCRAMBLEPROBLEM\_H

#include <iostream>

#include <string>

#include <cstdlib>

#include <ctime>

#include "quizItem.h"

using namespace std;

class WordScrambleProblem : public QuizItem

{

private:

// DATA MEMBERS

string original;

public:

// SETS

void setWord(string change)

{

original = change;

}

// GETS

string getWord() const

{

return original;

}

// MEMBER FUNCTIONS

WordScrambleProblem();

void generateQuestion();

};

#endif

// Word Scramble Problem

// Author James

#include "wordscramble.h"

using namespace std;

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

// Constructor

//

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

WordScrambleProblem::WordScrambleProblem()

{

}

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

// Virtual Word Scramble Problem

//

// Recives an array of words

// Finds one and returns 1 scrambled and 1 unscrambled

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

void WordScrambleProblem::generateQuestion()

{

// Initilize

string original = "monitor", temp, scramble = "";

unsigned int i, jump;

int newIndex[100];

int goodData, select;

/\*

// Generate 1 random number 0-GoodData

// Find random trivia question

select = rand() % goodData;

// Set random comp words question and answer

original = compWords[select];

\*/

// Assign word to temp

temp = original;

for (i = 0; i < original.length(); i++)

{

// mark all as un modified

newIndex[i] = -1;

}

// Step through word at random

for (i = 0; i < original.length(); i++)

{

// Find random letter in word

jump = rand() % original.length();

// Finds unscrambled letters of the word

while (newIndex[jump] != -1)

{

// Increment letters by 1

jump++;

// If the jump is bigger than the number of letters start at 0

if (jump >= original.length())

{

jump = 0;

}

}

// Assign letter to a new spot

newIndex[jump] = i;

}

// Assign letters to their spots in the word

for (i = 0; i < original.length(); i++)

{

scramble += temp[newIndex[i]];

}

// Set the question and answer in quiz item

setQuestion(scramble);

setAnswer(original);

}

// Word Scramble Problem

// Author James Wetters

#ifndef TRIVIAPROBLEM\_H

#define TRIVIAPROBLEM\_H

#include <iostream>

#include <string>

#include <cstdlib>

#include <ctime>

#include "quizItem.h"

using namespace std;

// Constants

const int MAXARRAY = 10;

class TriviaProblem : public QuizItem

{

private:

// DATA MEMBERS

string triviaQuestion, triviaAnswer;

public:

// SETS

void setTriviaQuestion(string change)

{

triviaQuestion = change;

}

void setTriviaAnswer(string change)

{

triviaAnswer = change;

}

// GETS

string getTriviaQuestion() const

{

return triviaQuestion;

}

string getTriviaAnswer() const

{

return triviaAnswer;

}

// MEMBER FUNCTIONS

TriviaProblem();

void generateQuestion();

};

#endif

// Word Scramble Problem

// Author James Wetters

#include "trivia.h"

using namespace std;

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

// Constructor

//

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

TriviaProblem::TriviaProblem()

{

}

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

// Virtual Generate Question Trivia Problem

//

// Recives a 2d array of triva questions and answers

// Sends 1 trivia question back and 1 answer

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

void TriviaProblem::generateQuestion()

{

// Initilize

string triviaQuestion, triviaAnswer;

int goodData, select;

/\*

// Generate 1 random number 0-GoodData

// Find random trivia question

select = rand() % goodData;

// Set random trivia question and answer

triviaQuestion = trivia[select][0];

triviaAnswer = trivia[select][1];

\*/

// Place holder

triviaQuestion = "What continent has the fewest flowering plants ?";

triviaAnswer = "Antarctica";

// Set question and answer

setQuestion(triviaQuestion);

setAnswer(triviaAnswer);

}