// Some Sample Code to help in improving your solutions in the Week 3 area of the course. I did this

// using Eclipse environment.

//Week 3 Sample Solution to build on in Week 4

//CS 215 Fall 2015

//Dr. Howard Evans

**#include** <iostream>

**#include** <string>

**#include** <fstream>

**#include** <cstdlib>

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

/\* Class files for the Question Classes. Designed so that the base class could be used as the collection class later in the program. Base Question Class has several methods that are identified as virtual as they will be redefined later in the hierarchy to specify the types of questions that will be read from or written to files \*/

**class** Question // super class

{

**public**:

string **getQuestion**()//gets the question

{

**return** question;

}

**virtual** **int** **getValue**() //gets the point value of the question

{

**return** value;

}

**virtual** string **getQuestionType**()// gets the type of question

{

**return** questiontype;

}

**virtual** **void** **setQuestion**(string answer, **int** value)

{

}

**virtual** **void** **setNewQuestion**(string answer, **int** value)

{

}

**virtual** **void** **printOptions**()

{

}

**virtual** string **getAnswer**()

{

**return** answer;

}

**private**:

string question, answer;

**int** value;

string questiontype;

};

//The class for True/False Questions

**class** QuestionTF: **public** Question// class for true and false questions

{

**public**:

**void** **setQuestion**(string theQuestion, **int** pointValue)

{

string theAnswer;

questiontype = "TF";

question = theQuestion;

value = pointValue;

options = "true/false";

//get the answer from the file

getline(cin,theAnswer);

answer = theAnswer;

}

**void** **setNewQuestion**(string theQuestion, **int** pointValue)

{

string theAnswer;

questiontype = "TF";

question = theQuestion;

value = pointValue;

options = "true/false";

//get the answer from user

cout<<"Enter answer true/false\n";

getline(cin,theAnswer);

answer = theAnswer;

}

**int** **getValue**() //gets the point value of the question

{

**return** value;

}

string **getQuestionType**()// gets the type of question

{

**return** questiontype;

}

**void** **printOptions**()//prints the options for that question

{

cout<<question<<endl;

cout<<answer<<endl;

}

string **getAnswer**()//outputs the answer for that question

{

**return** answer;

}

**private**:

string question, questiontype;

string answer;

string options;

**int** value;

};

//The class for Multiple Choice Questions

**class** QuestionMC: **public** Question//class for multiple choice

{

**public**:

**void** **setQuestion**(string theQuestion, **int** pointValue)

{

string line;

questiontype = "MC";

//get the number of choices from the file

getline(cin,line);

numberOfOptions = **atoi**(line.c\_str());

question = theQuestion;

value = pointValue;

//get the individual choice lines and load to options array

**for** (**int** count = 0; count<numberOfOptions;count++){

getline(cin,line);

options[count] = line;

}

//get the answer from the file and load into answer

getline(cin, line);

answer = line;

}

**void** **setNewQuestion**(string theQuestion, **int** pointValue)

{

string line;

questiontype = "MC";

//get the number of choices from the user

cout<<"Enter the number of choices: ";

getline(cin,line);

numberOfOptions = **atoi**(line.c\_str());

question = theQuestion;

value = pointValue;

//get the individual choice lines and load to options array

**for** (**int** count = 0; count<numberOfOptions;count++){

cout<<"\nEnter next option: ";

getline(cin,line);

options[count] = line;

}

//get the answer from the user and load into answer

cout<<"\nEnter Answer: ";

getline(cin, line);

answer = line;

}

**void** **printOptions**()// prints the questions, options, and answer

{

**char** first = 'A';

cout<<question<<endl;

**for**(**int** count = 0; count<numberOfOptions;count++){

cout<<first++ <<". "<<options[count]<<"\n";

}

cout<< answer << "\n";

}

**int** **getValue**() //gets the point value of the question

{

**return** value;

}

string **getQuestionType**()// gets the type of question

{

**return** questiontype;

}

string **getAnswer**()// prints the answer

{

**return** answer;

}

**private**:

**int** numberOfOptions;

string question, answer;

string options[6];

string questiontype;

**int** value;

};

// Function prototypes for the functions supporting the main program functionality

**int** **loadArray**(Question \*myQuestions[]);

**int** **addQuestion**(Question \*myQuestions[], **int** numquestions);

**void** **printQuizQuestions**(Question \*myQuestions[], **int** numquestions);

**void** **writeExamQuestionFile**(Question \*myQuestions[], **int** numquestions);

/\* this is something to write the test bank test file from the text file given in Week 2 as the starting point. Implemented some changes that allow you to read from this file like it was cin and write like it was cout by redirecting the read and write streams in the program. Somewhat simplifies the file handling. You might use a text file that you create separately and avoid this for Week 2. Week 3 might be more focused on the Add Question functionality. At the beginning of this program we load the question array from an input file and at the end of this program we write to an exam file. \*/

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

Question \*myQuestions[10];

**int** numquestions;

//opening the testbank file and processing as a question of each type

**try** {

ifstream infile("testbank2.txt");

streambuf \*cinbuf = cin.rdbuf(); //save old buf

cin.rdbuf(infile.rdbuf()); //redirect std::cin to infile.txt!

string line, theQuestion, theAnswer;

numquestions = loadArray(myQuestions);

cin.rdbuf(cinbuf); //reset to standard input again

}

**catch** (exception& e) {

cout << "Hey man, you tried to get away with dividing by zero on line 204, so I jumped to the catch part with the exceptions!!!";

cout << "Whoa buddy!!! Everything stops here!!! Something went wrong with reading the infile like: " << e.**what**() << ". Check the contents of testbank2.txt";

}

numquestions = addQuestion(myQuestions, numquestions);

printQuizQuestions(myQuestions, numquestions);

cout<<"Write to exam File";

**try** {

// code here

ofstream outfile("exam.txt");

streambuf \*coutbuf = std::cout.rdbuf(); //save old buf

cout.rdbuf(outfile.rdbuf()); //redirect std::cout to out.txt!

writeExamQuestionFile(myQuestions, numquestions);

cout.rdbuf(coutbuf); //reset to standard output again

cout<<"Exam File Written.\n"<<"\n";

}

**catch** (exception& e) {

cout << "Exception occurred";

}

**getchar**();

**return** 0;

}

// Function to load the array of Questions from the input file

**int** **loadArray**(Question \*myQuestions[])

{

string line;

string questiontype, theQuestion;

**int** numquestions, questionvalue;

//get the number of questions from the first line in the file

getline(cin,line);

numquestions = **atoi**(line.c\_str());

**for**(**int** count = 0; count<numquestions;count++){

getline(cin,line);

//get the next line with the question type and the value of the question

**int** npos = line.size();

**int** prev\_pos = 0;

**int** pos = 0;

**while**( line[pos]!=' ')

pos++;

questiontype = line.substr(prev\_pos, pos-prev\_pos);

prev\_pos = ++pos;

questionvalue = **atoi**(line.substr(prev\_pos, npos-prev\_pos).c\_str()); // Last word

//process a true/false question

**if** (questiontype == "TF")

{

myQuestions[count] = **new** QuestionTF;

getline(cin,theQuestion);

myQuestions[count]->setQuestion(theQuestion,questionvalue);

}

//process a multiple choice question

**if** (questiontype =="MC")

{

myQuestions[count] =**new** QuestionMC;

getline(cin,theQuestion);

myQuestions[count]->setQuestion(theQuestion,questionvalue);

}

}

**return** numquestions;

}

//Function to add questions by getting from user and using he interactive methods //added to the classes.

**int** **addQuestion**(Question \*myQuestions[], **int** numquestions)

{

**int** questionvalue;

**int** count=numquestions;

string theQuestion,line, questiontype;

cout<<"Enter the Question type and value\n";

getline(cin,line);

//get the next line with the question type and the value of the question

**int** npos = line.size();

**int** prev\_pos = 0;

**int** pos = 0;

**while**( line[pos]!=' ')

pos++;

questiontype = line.substr(prev\_pos, pos-prev\_pos);

prev\_pos = ++pos;

questionvalue = **atoi**(line.substr(prev\_pos, npos-prev\_pos).c\_str()); // Last word

//process a true/false question

**if** (questiontype == "TF")

{

myQuestions[count] = **new** QuestionTF;

cout<<" \nEnter the Question: ";

getline(cin,theQuestion);

myQuestions[count]->setNewQuestion(theQuestion,questionvalue);

}

//process a multiple choice question

**if** (questiontype =="MC")

{

myQuestions[count] =**new** QuestionMC;

cout<<" \nEnter the Question: ";

getline(cin,theQuestion);

myQuestions[count]->setNewQuestion(theQuestion,questionvalue);

}

**return** ++numquestions;

}

// Function to print out the Quiz questions from the Array. Uses methods from the

// classes developed to allow this printing to be done generically to screen or file.

**void** **printQuizQuestions**(Question \*myQuestions[], **int** numquestions)

{

//print out the questions that have been processed

**for**(**int** count = 0; count<numquestions;count++)

{

myQuestions[count]->printOptions();

cout<<"\n";

}

}

//Function to write to the file in the format needed to read later when loading if //needed.

**void** **writeExamQuestionFile**(Question \*myQuestions[], **int** numquestions)

{

**int** count;

string qtype;

cout<<numquestions<<"\n";

**for** (count=0;count<numquestions;count++){

qtype=myQuestions[count]->getQuestionType();

cout<<qtype<<" "<<myQuestions[count]->getValue()<<"\n";

myQuestions[count]->printOptions();

}

}