Jun Kim

Satish Singhal

CSCS-1 #0105

May. 27, 2021

Assignment 8: IOAA Document

**\* IOAA form struct Quiz**

Struct Members

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Remarks |
| question | string | stores the question part of quiz |
| key | char | stores the answer key to the question |
| response | char | stores the user response to the question |

Pseudocode representation of struct

struct Quiz

question

key

response

end struct

**\* IOAA for main function**

Function Input through Argument

None

User Input During Function Execution

None

Global Constants

|  |  |  |  |
| --- | --- | --- | --- |
| Variable name | C++ Data Type | Value | Remarks |
| MAX | int | 20 | maximum capacity of arrays – question array, answer array, and response array |

Function Output

Function calls other functions as illustrated in algorithm below

Function outputs to console the score and percentage obtained in test

Function also outputs whether user passed or failed the test.

Analysis

percent = (score/number of questions) \* 100

Algorithm

1. Add all the #include directives, declare global constants
2. Quiz qz[MAX]
3. ifstream input
4. Print, “Welcome to driving test.”, EOL
5. openInputFile(input, “questions”)
6. int lenQ = fillQuestions (qz, input)
7. input.close()
8. openInputFile(input, “answers”)
9. int lenA = fillAnswerKeys(qz, input)
10. input.close()
11. if (lenQ != lenA) then
    1. Print, “Number of questions and answers are different.”, EOL
12. else
    1. fillResponses(qz, lenQ)
    2. int score = gradeResponses(qz, lenQ)
    3. Print, EOL, “You scored “, score, “ points out of total “, lenQ, “ points.”, EOL
    4. double percent = (static\_cast<double>(score)/lenQ) \* 100
    5. if (percent >= 60.0) then
       1. Print, “You passed exam. Congratulation!”, EOL
    6. else
       1. Print, “You failed. Please review your answers.”, EOL
    7. end if
    8. Print, “Total answered correctly: “, score, EOL
    9. Print, “Total answered incorrectly: “, (lenQ – score), EOL
    10. printAnswersAndResponses(qz, lenQ)
13. end if

**\* IOAA for function void openInputFile(ifstream & input, const string & message)**

Purpose of the function

Asks user for the path to input files (which could be questions or answer key files), and opens either, and bonds these files to ifstream object that is passed by reference. the openFile function manages so that either file is opened successfully, file exists, and file has dome data in it.

Function Input through Arguments

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| input | ifstream | ifstream object passed from main function not attached to an input file |
| message | string | would have a value “questions” for questions file and “answers” for answers file |

User Input During Function Excution

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| In\_File | string | file name either for questions file (if message values is questions) or for answers file |

Function Output

Function returns to main function by reference the ifstream object bonded to input file

Analysis

None

Algorithm

1. bool done = false
2. declare In\_File as string and initialization with “”
3. while (!done)
   1. input.clear()
   2. Print, “Please input the name of the data file with “, message, “ to be read: “, EOL
   3. getline(cin, In\_File)
   4. Print, “The file name entered is: “, EOL, In\_File, EOL
   5. input.open(In\_File)
   6. if (input.fail()) then
      1. Print, “The file “, In\_File, “ does not exist.”, EOL
      2. done = false
   7. else
      1. input.peek()
      2. if (input.eof()) then
         1. Print, “The file has no data in it.”, EOL
         2. done = false
         3. input.close()
      3. else
         1. done = true
      4. end if
   8. end if
4. end while
5. Print, “File “, In\_File, “ opened and has data in it.”, EOL, EOL
6. end of function

**\* IOAA for function int fillQuestions (Quiz qz[], ifstream & in)**

Purpose of the function

Reads questions from the question file and fills an array with the questions, so that there is one question per array element in the question field. Function takes an empty questions array and ifstream object bonded to the question file. Functions fills the qz array with questions and returns the logical array length by return mechanism.

Function Input through Argument

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| qz[] | Quiz | empty array of Quiz struct |
| in | ifstream | ifstream object bonded to questions file that has been already opened |

User Input During Function Execution

None

Function Output

Function returns to main function by reference the filled array called qz[] and ifstream object in passed to it by reference. The question fields of each array member are populated by the functions as read from the file.

Analysis

None

Algorithm

1. declare count as int and initialization with 0 (zero)
2. while (in.peek() != EOF and count < MAX)
   1. declare temp as string and initialization with “”
   2. in.ignore()
   3. getline(in, temp, ‘$’)
   4. qz[count].question = temp
   5. count = count + 1
3. end while
4. return count
5. end of function

**\* IOAA for function int fillAnswerKeys(Quiz qz[], ifstream & inA)**

Purpose of the function

Reads answers from the answers file and fills qz array with the answer keys by populating the key field of each array member. Function takes the qz array and ifstream object bonded to the answer file. Function after filling the qz array with answer keys and returns the logical array length by return mechanism.

Function Input through Argument

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| qz[] | Quiz | empty array of Quiz struct |
| inA | ifstream | ifstream object bonded to answers file that has been already opened |

User Input During Function Execution

None

Function Output

Function returns to main function by reference the filled array called qz[] and ifstream object in passed to it by reference. The answer fields of each array member are populated by the functions as read from the file.

Analysis

None

Algorithm

1. declare count as int and initialization with 0 (zero)
2. while (inA.peek() != EOF and count<MAX)
   1. declare temp as char and initialization with ‘ ‘
   2. temp <- inA
   3. qz[count].key = temp
   4. count = count + 1
3. end while
4. return count
5. end of function

**\* IOAA for function void fillResponse (Quiz qz[], int len)**

Purpose of the function

The function manages that presenting the user the questions, getting user response to each question and storing them in the response field of the qz array. The function takes as arguments, the partially filled qz array, and its logical length.

Function Input through Argument

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| qz[] | Quiz | filled array containing questions but empty spaces for response field |
| len | int | logical length of qz[] array |

User Input During Function Execution

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| temp | char | user will input answer either A, B, C, or D depending on the quiz |

Function Output

Function returns to main function by reference the qz[] array in which user response is recorded in each struct member in the response field.

Analysis

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| flag\_Res | bool | true: res == ‘A’ or res == ‘B’ or res == ‘C’ or res == ‘D’  false: ifelse |

Algorithm

1. Print, “ \* \* \* Driver’s License Practice Exam \* \* \*”, EOL
2. Print, “Please answer the questions as asked. Questions are multiple choice.”, EOL
3. Print, “Enter correct letter response out of A, B, C, or D.”, EOL, EOL
4. for (int index=0, index < len, index = index + 1)
   1. Print, qz[index].question, EOL
   2. declare temp as char and initialization with ‘ ‘
   3. declare flag\_Res as bool and initialization with false
   4. do
      1. Print, “Please choose your answer: “
      2. temp<-
      3. temp = toupper(temp)
      4. if (temp == ‘A’ or temp == ‘B’ or temp == ‘C’ or temp == ‘D’) then
         1. flag\_Res = true
      5. else
         1. Print, “You entered wrong value. Please, choose correct one.”, EOL
      6. end if
   5. while (flag\_Res == false)
   6. end do-while loop
5. qz[index].response = temp
6. end for
7. end of function

**\* IOAA for function int gradeResponses(const Quiz qz[], int len)**

Purpose of the function

The function compares the answer key and responses field of same member of qz array and returns the correct number of responses by return mechanism. Function also takes len as argument – the logical length of both arrays.

Function Input through Argument

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| qz[] | Quiz | filled array containing answers, and responses for each struct member of array |
| len | int | logical length of qz[] array |

User Input During Function Execution

None

Function Output

Function returns to caller function by a return mechanism the correct number of answers by the user in the quiz

Analysis

None

Algorithm

1. declare score as int and initialization with 0 (zero)
2. for (int index = 0, index < len, index = index + 1)
   1. if (qz[index].key == qz[index].response) then
      1. score = score + 1
   2. end if
3. end for
4. return score
5. end of function

**\* IOAA for function void printAnswersAndResponses(const Quiz qz[], int len)**

Purpose of the function

The function prints questions, correct answers, and user responses.

Function Input through Argument

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| qz[] | Quiz | filled array containing questions, answers, and responses for each struct member of array |
| len | int | logical length of qz[] array |

User Input During Function Execution

None

Function Output

Function outputs to console (for user feedback), the questions, correct answers, user answers

Analysis

None

Algorithm

1. Print, EOL, “Now we give summary of all questions, answers and your responses.”, EOL
2. for (int index = 0, index < len, inex = index + 1)
   1. Print, qz[index].question, EOL
   2. Print, “Correct answer = “, qz[index].key, EOL
   3. Print, “Your answer = “, qz[index].response, EOL
   4. if (qz[index].key == qz[index].response) then
      1. Print, “You got this question right.”, EOL
   5. else
      1. Print, “You got this question wrong.”, EOL
   6. end if
   7. if (index == (len-1)) then
      1. Print, “Thank you for using driving test”, EOL
      2. Print, “Goodbye!”, EOL
   8. else
      1. Print, “Press any key to continue…”, EOL
      2. cin.get() //Windows needs to be used “system(“pause”)
   9. end if
3. end for
4. end of function