Jun Kim

Satish Singhal

CSCS-1 #0105

May. 18, 2021

Assignment 7: IOAA Document

**\* IOAA for main function**

Function Input through Argument

None

User Input During Function Execution

None

Global Constants

|  |  |  |  |
| --- | --- | --- | --- |
| Variable name | C++ Data Type | Value | Remark |
| MAX | size\_t | 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. Print, “Welcome to driving test.”, EOL
3. ifstream in
4. openFile(in, “questions”)
5. string questions [MAX]
6. int lenQ = fillQuestionsArray(questions, in)
7. in.close()
8. openFile(in, “answers”)
9. char ans [MAX]
10. int lenA = fillAnswerKeyArray(ans, in)
11. if (lenQ != lenA) then
    1. Print, “Number of questions and answers are different.”, EOL
12. else
    1. char response [MAX]
    2. int lenR = fillResponseArray(questions, response, lenQ)
    3. declare score as int and put initial value 0 (zero)
    4. if (lenQ != lenR) then
       1. Print, “Number of questions and responses are different,”, EOL
    5. else
       1. score = GradeResponses(ans, reponse, lenQ)
       2. Print, “You scored “, score, “ points out of total “, lenQ, “ points.”, EOL
       3. double percent = (static\_cast<double>(score)/lenQ) \* 100
       4. if (percent >= 60.0) then
          1. Print, “You passed exam. Congratulation!”, EOL
       5. else
          1. Print, “You failed. Please review your answers.”, EOL
       6. end if
       7. Print, “Total answered correctly: “, score, EOL
       8. Print, “Total answered incorrectly: “, (lenQ – score), EOL
    6. end if
    7. printAnswersAndResponses(questions, ans, response, lenQ)
13. end if
14. in.close()

**\* IOAA for function openFile(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: “, In\_File, EOL
   5. input.open(In\_File)
   6. if (input.fail()) then
      1. Print, “The file “, In\_File, “ does not exist.”, EOL
      2. continue
   7. else
      1. if (input.peek() == EOF) then
         1. Print, “The file has no data in it.”, EOL
         2. input.close()
         3. continue
      2. else
         1. done = true
      3. end if
   8. end if
4. end while
5. Print, “File “, In\_File, “ opened and has data in it.”, EOL
6. end of function

**\* IOAA for function int fillQuestionsArray(string questions[], 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. Function takes an empty questions array and ifstream object bonded to the file. Functions fills the questions array with questions and returns the logical array length by return mechanism.

Function Input through Argument

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

User Input During Function Execution

None

Analysis

None

Algorithm

1. declare i as int and initialization with 0 (zero)
2. while (in.peek() != EOF and I < MAX)
   1. declare question as string and initialization with “”
   2. declare ch as char and initialization with ‘ ‘
   3. while (in.peek() != ‘$’) //Same function with “getline(in, question, ‘$’)”
      1. ch = in.get()
      2. question = question + ch
   4. end while
   5. questions[i] = question
   6. in.ignore(256, ‘\n’)
   7. i = i + 1
3. end while
4. return i

**\* IOAA for function int fillAnswerKeyArray(char ans[], ifstream & in)**

Purpose of the function

Reads answers from the answers file and fills an array with the answer keys, so that there is one answer key per array element. Function takes an empty answers array and ifstream object bonded to the file. Function fills the answers array with answers and returns the logical array length by return mechanism.

Function Input through Argument

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| ans[] | char | empty array of char to contain answers in the answer file |
| in | 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 answers and

ifstream object in passed to it by reference

Analysis

None

Algorithm

1. declare i as int and initialization with 0 (zero)
2. while (in.peek() != EOF and i<MAX)
   1. ans[i] <- in
   2. i = i + 1
3. end while
4. return i

**\* IOAA for function int fill ResponseArray(const string questions[], char response[], 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 a response array, and keeping track of logical length of the response array. The function records user response to the questions stored in questions array and returns the response array by reference and returns logical length of response array by return mechanism.

Function Input through Argument

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| questions[] | string | filled array containing questions for the quiz |
| response[] | char | unfilled array which will record user response entered as A, B, C, or D |
| len | int | logical length of questions and responses array |

User Input During Function Execution

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| res | 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 filled response array that is populated by the user response

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. declare res as char and initialization with ‘ ‘
5. declare i as int and initialization with 0 (zero)
6. for (i, i < len, i = i + 1)
   1. Print, questions[i], EOL
   2. declare flag\_Res as bool and initialization with false
   3. do
      1. Print, “Please choose your answer: “
      2. res<-
      3. res = toupper(res)
      4. if (res == ‘A’ or res == ‘B’ or res == ‘C’ or res == ‘D’) then
         1. flag\_Res = true
      5. else
         1. Print, “You entered wrong value. Please, choose correct one.”, EOL
   4. while (flag\_Res == false)
   5. end do-while loop
7. response[i] = res
8. end for
9. return i

**\* IOAA for function int GradeResponses(const char answers[], const char response[], int len)**

Purpose of the function

The function compares the answer key and responses array and returns the correct number of responses by return mechanism.

Function Input through Argument

|  |  |  |
| --- | --- | --- |
| Variable name | C++ Data Type | Remarks/Comments |
| answers[] | char | filled array containing answer key values for the quiz |
| response[] | char | response array already filled with user response |
| len | int | logical length of answers and responses array |

User Input During Function Execution

None

Function Output

Function returns to main 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 (i = 0, i < len, i = i + 1)
   1. if (answers[i] == response[i]) then
      1. score = score + 1
   2. end if
3. end for
4. return score

**\* IOAA for function void printAnswersAndResponses(const string questions[], const char answers[], const char response[], 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 |
| questions[] | string | filled array containing questions for the quiz |
| answers[] | char | filled array containing answer key values for the quiz |
| response[] | char | response array already filled with user response |
| len | int | logical length of questions, answers and responses 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. declare i as int and initialization with 0 (zero)
3. for (i = 0, i < len, i = i + 1)
   1. Print, questions[i], EOL
   2. Print, “Correct answer = “, answers[i], EOL
   3. Print, “Your answer = “, response[i], EOL
   4. if (answers[i] == response[i]) then
      1. Print, “You got this question right.”, EOL
   5. else
      1. Print, “You got this question wrong.”, EOL
   6. end if
   7. if (i == (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
4. end for
5. end function