Consolidated Assignment 3 Report

This report contains the graded results for the newest of each exercise submitted to the assignment checker prior to 7/29/2020 5:10:17 PM PDT.

Student Name: Shaun Chemplavil
Student ID: U08713628
Contact e-mail: shaun.chemplavil@gmail.com
C/C++ Programming II (Section 149123)

Submitted:
```
   Exercise 1: 7/25/2020 9:25:57 AM PDT
   Exercise 2: 7/24/2020 3:52:16 PM PDT
   Exercise 3: 7/23/2020 7:44:16 AM PDT
   Exercise 4: 7/27/2020 3:57:44 PM PDT
```

A "for" statement (note 3.5) is typically the most appropriate choice whenever an object/variable meets at least two of the following criteria:

1. It can be initialized in the loop's "initial expression";
2. It is tested before entering the body of the loop;
3. It is updated after each execution of the loop body.

Score (out of 20 possible): 19.4

```
THIS WAS SENT FROM A NOTIFICATION-ONLY ADDRESS THAT CANNOT ACCEPT INCOMING MAIL.
For help please contact the instructor at the email address provided on the "Home" page
of the course's Canvas website.  The assignment checker DOES NOT GRADE your submissions
but merely reports on issues so you can correct them and resubmit, thereby avoiding
unnecessary credit loss.  ALL GRADING IS DONE MANUALLY BY THE INSTRUCTOR after the
assignment deadline based solely upon the NEWEST submission of each exercise.  BE WARY
of correcting minor issues after the deadline because a late deduction will usually be
much greater than a minor issue deduction.


   From: Shaun Chemplavil <mailto:shaun.chemplavil@gmail.com>
   Subject: C2A3E1_U08713628
   Submitted: 7/25/2020 9:25:57 AM PDT
   Course: C/C++ Programming II (Section 149123)
   Student's name: Shaun Chemplavil
   Contact email: shaun.chemplavil@gmail.com
   Student ID: U08713628
   Assignment 3, Exercise 1  (C2_005306999M02005X80306)
   Exercise point value: 2
   File submitted:
      C2A3E1_Sentences.txt

Your submission has been accepted and will be graded manually by the instructor.  You
may resubmit it as many times as you wish before the assignment deadline.  BE WARY of
correcting minor issues after the deadline because a late deduction will usually be
much greater than a minor issue deduction.
```

Shaun Chemplavil U08713628
shaun.chemplavil@gmail.com
C/C++ Programming II : Dynamic Memory and File I/O Concepts
149123 Raymond L. Mitchell, Jr., M.S.
07/25/2020
C2A3E1_Sentences.txt
Answers to Exercise testing Right-Left Rule and Decayed Right-Left Rule

1) fish decays to a pointer to a double
2) fish decays to a pointer to a double
3) fish decays to a pointer to a double
4) fish is an array of 57 doubles
~~5) fish is a double~~
6) fish is an array of 57 doubles
7) fish is an array of 57 doubles
~~8) fish is a double~~
~~9) fish is a double~~
10) fish decays to a pointer to a double

-0.6

THIS WAS SENT FROM A NOTIFICATION-ONLY ADDRESS THAT CANNOT ACCEPT INCOMING MAIL. For help please contact the instructor at the email address provided on the "Home" page of the course's Canvas website. The assignment checker DOES NOT GRADE your submissions but merely reports on issues so you can correct them and resubmit, thereby avoiding unnecessary credit loss. ALL GRADING IS DONE MANUALLY BY THE INSTRUCTOR after the assignment deadline based solely upon the NEWEST submission of each exercise. BE WARY of correcting minor issues after the deadline because a late deduction will usually be much greater than a minor issue deduction.

```
   From: Shaun Chemplavil <mailto:shaun.chemplavil@gmail.com>
   Subject: C2A3E2_U08713628
   Submitted: 7/24/2020 3:52:16 PM PDT
   Course: C/C++ Programming II (Section 149123)
   Student's name: Shaun Chemplavil
   Contact email: shaun.chemplavil@gmail.com
   Student ID: U08713628
   Assignment 3, Exercise 2  (C2_005477401M02005X2477)
   Exercise point value: 4
   Files submitted:
      C2A3E2_main-Driver.cpp
      C2A3E2_TestDeclarations.cpp

"Compile-time" results:
   No "compile-time" issues;
"Run-time" results:
   Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);
```

```
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C / C++ Programming II : Dynamic Memory and File I / O Concepts
// 149123 Raymond L.Mitchell, Jr., M.S.
// 07 / 24 / 2020
// C2A3E2_TestDeclarations.cpp
// Win10
// Visual C++ 19.0
//
// File containing definition of the TestDeclarations function, it contains
// five single statements to test understanding of "Reverse Right-Left" rule
//

const int ARRAY_SIZE = 6;

void TestDeclarations()
{
   long **(*afe)[ARRAY_SIZE];                    //1
   float (*pv)(int(*pa)[ARRAY_SIZE]) = 0;        //2
   afe = (long **(*)[ARRAY_SIZE])pv;             //3
   int &rF1(double *precision);                  //4
   int *rF3(double &precision);                  //5
}
```

```
********** C2 ASSIGNMENT 3 EXERCISE 2 AUTOMATIC PROGRAM RUN RESULTS **********

************* THE RESULTS BELOW HAVE BEEN PARTIALLY CHECKED AND **************
*************   NO ERRORS WERE FOUND.  HOWEVER, THIS DOES NOT   **************
*************  NECESSARILY MEAN THAT THERE ARE NO ERRORS.  THE  **************
*************  INSTRUCTOR WILL DO A MORE THOROUGH CHECK DURING  **************
*************                  MANUAL GRADING.                  **************


-------------------------------- START OF RUN --------------------------------

Assignment 3 Exercise 2 Complete!

--------------------------------- END OF RUN ---------------------------------
```

```
THIS WAS SENT FROM A NOTIFICATION-ONLY ADDRESS THAT CANNOT ACCEPT INCOMING MAIL.
For help please contact the instructor at the email address provided on the "Home" page
of the course's Canvas website.  The assignment checker DOES NOT GRADE your submissions
but merely reports on issues so you can correct them and resubmit, thereby avoiding
unnecessary credit loss.  ALL GRADING IS DONE MANUALLY BY THE INSTRUCTOR after the
assignment deadline based solely upon the NEWEST submission of each exercise.  BE WARY
of correcting minor issues after the deadline because a late deduction will usually be
much greater than a minor issue deduction.


   From: Shaun Chemplavil <mailto:shaun.chemplavil@gmail.com>
   Subject: C2A3E3_U08713628
   Submitted: 7/23/2020 7:44:16 AM PDT
   Course: C/C++ Programming II (Section 149123)
   Student's name: Shaun Chemplavil
   Contact email: shaun.chemplavil@gmail.com
   Student ID: U08713628
   Assignment 3, Exercise 3  (C2_001241579M02005X54241)
   Exercise point value: 6
   Files submitted:
      C2A3E3_RecordOpinions.c
      C2A3E3_main-Driver.c

"Compile-time" results:
   No "compile-time" issues;
"Run-time" results:
   Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);
```

```
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C / C++ Programming II : Dynamic Memory and File I / O Concepts
// 149123 Raymond L.Mitchell, Jr., M.S.
// 07 / 23 / 2020
// C2A3E3_RecordOpinions.c
// Win10
// Visual C++ 19.0
//
// File contains RecordOpinions function, which prompts users for ratings of
// a product and summarizes the total quantity of each rating in a table
//

#define ENDPOINT 10
#define TERMINATE 15
#define TOTAL_RATINGS (2 * ENDPOINT + 1)

#include <stdio.h>

void RecordOpinions(void)
{
   int userResp = 0;
   // Initial Array to hold user rating
   // (rawArray[0] refers to rating of -ENDPOINT)
   int rawArray[TOTAL_RATINGS] = {0};
   // Pointer to access rawArray in by merely using user rating as an index
   // (ratingCounters[0] refers to rating of 0)
   int *ratingCounters = &rawArray[ENDPOINT];

   // prompt user for valid rating until 'TERMINATE' value is entered
   while (userResp != TERMINATE)
   {
      // get user rating
      printf("Enter your rating: (valid range -%d to %d, to end enter %d): ",
         ENDPOINT, ENDPOINT, TERMINATE);
      scanf("%d", &userResp);

      // increment count at response table index
      if (userResp >= -ENDPOINT && userResp <= ENDPOINT)
         ratingCounters[userResp]++;
      // Invalid Entry, prompt for new rating
      else if (userResp != TERMINATE)
         printf("Rating out of range, Try again!\n");
   }

   // Display table summary of ratings
   printf(
      "Rating  Responses\n"
      "------  ----------\n");
   for (int rating = -ENDPOINT; rating <= ENDPOINT; rating++)
      printf("%5d    %4d\n", rating, ratingCounters[rating]);
}
```

```
********** C2 ASSIGNMENT 3 EXERCISE 3 AUTOMATIC PROGRAM RUN RESULTS **********

************* THE RESULTS BELOW HAVE BEEN PARTIALLY CHECKED AND **************
*************   NO ERRORS WERE FOUND.  HOWEVER, THIS DOES NOT   **************
*************  NECESSARILY MEAN THAT THERE ARE NO ERRORS.  THE  **************
*************  INSTRUCTOR WILL DO A MORE THOROUGH CHECK DURING  **************
*************                  MANUAL GRADING.                  **************


-------------------------- CODE CHANGES FOR 1ST RUN --------------------------
ENDPOINT = 2 & TERMINATE = -100
------------------------------ START OF 1ST RUN ------------------------------

Enter your rating: (valid range -2 to 2, to end enter -100): -7
Rating out of range, Try again!
Enter your rating: (valid range -2 to 2, to end enter -100): 22
Rating out of range, Try again!
Enter your rating: (valid range -2 to 2, to end enter -100): 0
Enter your rating: (valid range -2 to 2, to end enter -100): 0
Enter your rating: (valid range -2 to 2, to end enter -100): 0
Enter your rating: (valid range -2 to 2, to end enter -100): 3
Rating out of range, Try again!
Enter your rating: (valid range -2 to 2, to end enter -100): 3
Rating out of range, Try again!
Enter your rating: (valid range -2 to 2, to end enter -100): 3
Rating out of range, Try again!
Enter your rating: (valid range -2 to 2, to end enter -100): 2
Enter your rating: (valid range -2 to 2, to end enter -100): -1
Enter your rating: (valid range -2 to 2, to end enter -100): -2
Enter your rating: (valid range -2 to 2, to end enter -100): -1
Enter your rating: (valid range -2 to 2, to end enter -100): 3
Rating out of range, Try again!
Enter your rating: (valid range -2 to 2, to end enter -100): -1
Enter your rating: (valid range -2 to 2, to end enter -100): -2
Enter your rating: (valid range -2 to 2, to end enter -100): -2
Enter your rating: (valid range -2 to 2, to end enter -100): -2
Enter your rating: (valid range -2 to 2, to end enter -100): -2
Enter your rating: (valid range -2 to 2, to end enter -100): 0
Enter your rating: (valid range -2 to 2, to end enter -100): -100
Rating  Responses
------  ----------
   -2       5
   -1       3
    0       4
    1       0
    2       1

------------------------------- END OF 1ST RUN -------------------------------

-------------------------- CODE CHANGES FOR 2ND RUN --------------------------
ENDPOINT = 0 & TERMINATE = 999
------------------------------ START OF 2ND RUN ------------------------------

Enter your rating: (valid range -0 to 0, to end enter 999): 1
Rating out of range, Try again!
Enter your rating: (valid range -0 to 0, to end enter 999): 0
Enter your rating: (valid range -0 to 0, to end enter 999): 2
Rating out of range, Try again!
Enter your rating: (valid range -0 to 0, to end enter 999): 0
Enter your rating: (valid range -0 to 0, to end enter 999): 0
Enter your rating: (valid range -0 to 0, to end enter 999): 0
Enter your rating: (valid range -0 to 0, to end enter 999): 0
Enter your rating: (valid range -0 to 0, to end enter 999): 0
Enter your rating: (valid range -0 to 0, to end enter 999): 0
Enter your rating: (valid range -0 to 0, to end enter 999): 0
Enter your rating: (valid range -0 to 0, to end enter 999): -999
Rating out of range, Try again!
Enter your rating: (valid range -0 to 0, to end enter 999): 0
Enter your rating: (valid range -0 to 0, to end enter 999): 0
Enter your rating: (valid range -0 to 0, to end enter 999): 0
Enter your rating: (valid range -0 to 0, to end enter 999): 999
Rating  Responses
------  ----------
    0      11

------------------------------- END OF 2ND RUN -------------------------------

-------------------------- CODE CHANGES FOR 3RD RUN --------------------------
ENDPOINT = 5 & TERMINATE = 128
------------------------------ START OF 3RD RUN ------------------------------

Enter your rating: (valid range -5 to 5, to end enter 128): -5
Enter your rating: (valid range -5 to 5, to end enter 128): -20
Rating out of range, Try again!
Enter your rating: (valid range -5 to 5, to end enter 128): 60
Rating out of range, Try again!
Enter your rating: (valid range -5 to 5, to end enter 128): -5
Enter your rating: (valid range -5 to 5, to end enter 128): 4
Enter your rating: (valid range -5 to 5, to end enter 128): 5
Enter your rating: (valid range -5 to 5, to end enter 128): 0
Enter your rating: (valid range -5 to 5, to end enter 128): 0
Enter your rating: (valid range -5 to 5, to end enter 128): 1
Enter your rating: (valid range -5 to 5, to end enter 128): 1
Enter your rating: (valid range -5 to 5, to end enter 128): 3
Enter your rating: (valid range -5 to 5, to end enter 128): 3
Enter your rating: (valid range -5 to 5, to end enter 128): -3
Enter your rating: (valid range -5 to 5, to end enter 128): 4
Enter your rating: (valid range -5 to 5, to end enter 128): 5
Enter your rating: (valid range -5 to 5, to end enter 128): 2
Enter your rating: (valid range -5 to 5, to end enter 128): 2
Enter your rating: (valid range -5 to 5, to end enter 128): -2
Enter your rating: (valid range -5 to 5, to end enter 128): -2
Enter your rating: (valid range -5 to 5, to end enter 128): -4
Enter your rating: (valid range -5 to 5, to end enter 128): 0
Enter your rating: (valid range -5 to 5, to end enter 128): 0
Enter your rating: (valid range -5 to 5, to end enter 128): 5
Enter your rating: (valid range -5 to 5, to end enter 128): -2
Enter your rating: (valid range -5 to 5, to end enter 128): -1
Enter your rating: (valid range -5 to 5, to end enter 128): -1
Enter your rating: (valid range -5 to 5, to end enter 128): -1
Enter your rating: (valid range -5 to 5, to end enter 128): 3
Enter your rating: (valid range -5 to 5, to end enter 128): 2
Enter your rating: (valid range -5 to 5, to end enter 128): 4
Enter your rating: (valid range -5 to 5, to end enter 128): 5
Enter your rating: (valid range -5 to 5, to end enter 128): 7
Rating out of range, Try again!
Enter your rating: (valid range -5 to 5, to end enter 128): 2
Enter your rating: (valid range -5 to 5, to end enter 128): 3
Enter your rating: (valid range -5 to 5, to end enter 128): 4
Enter your rating: (valid range -5 to 5, to end enter 128): 4
Enter your rating: (valid range -5 to 5, to end enter 128): 4
Enter your rating: (valid range -5 to 5, to end enter 128): 4
Enter your rating: (valid range -5 to 5, to end enter 128): 4
Enter your rating: (valid range -5 to 5, to end enter 128): 4
Enter your rating: (valid range -5 to 5, to end enter 128): 4
Enter your rating: (valid range -5 to 5, to end enter 128): -6
Rating out of range, Try again!
Enter your rating: (valid range -5 to 5, to end enter 128): 1
Enter your rating: (valid range -5 to 5, to end enter 128): 2
Enter your rating: (valid range -5 to 5, to end enter 128): 128
Rating  Responses
------  ----------
   -5       2
   -4       1
   -3       1
   -2       3
   -1       3
    0       4
    1       3
    2       5
    3       4
    4      10
    5       4

------------------------------- END OF 3RD RUN -------------------------------
```

```
THIS WAS SENT FROM A NOTIFICATION-ONLY ADDRESS THAT CANNOT ACCEPT INCOMING MAIL.
For help please contact the instructor at the email address provided on the "Home" page
of the course's Canvas website.  The assignment checker DOES NOT GRADE your submissions
but merely reports on issues so you can correct them and resubmit, thereby avoiding
unnecessary credit loss.  ALL GRADING IS DONE MANUALLY BY THE INSTRUCTOR after the
assignment deadline based solely upon the NEWEST submission of each exercise.  BE WARY
of correcting minor issues after the deadline because a late deduction will usually be
much greater than a minor issue deduction.


   From: Shaun Chemplavil <mailto:shaun.chemplavil@gmail.com>
   Subject: C2A3E4_U08713628
   Submitted: 7/27/2020 3:57:44 PM PDT
   Course: C/C++ Programming II (Section 149123)
   Student's name: Shaun Chemplavil
   Contact email: shaun.chemplavil@gmail.com
   Student ID: U08713628
   Assignment 3, Exercise 4  (C2_001880221M02005X40880)
   Exercise point value: 8
   Files submitted:
      C2A3E4_main-Driver.c
      C2A3E4_ParseStringFields.c
      C2A3E4_OpenFile.c

"Compile-time" results:
   No "compile-time" issues;
"Run-time" results:
   Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);
```

```c
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C / C++ Programming II : Dynamic Memory and File I / O Concepts
// 149123 Raymond L.Mitchell, Jr., M.S.
// 07 / 27 / 2020
// C2A3E4_OpenFile.c
// Win10
// Visual C++ 19.0
//
// File contains OpenFile function, which opens file located at the pointer
// passed to it
//

#include <stdio.h>
#include <stdlib.h>

FILE *OpenFile(const char *fileName)
{
   FILE *source;

   // open fileName in "read" mode, return error if failed
   if ((source = fopen(fileName, "r")) == NULL)
   {
      fprintf(stderr, "%s  failed to open \n", fileName);
      exit(EXIT_FAILURE);
   }

   return(source);
}
```

```
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C / C++ Programming II : Dynamic Memory and File I / O Concepts
// 149123 Raymond L.Mitchell, Jr., M.S.
// 07 / 27 / 2020
// C2A3E4_ParseStringFields.c
// Win10
// Visual C++ 19.0
//
// File contains ParseStringFields function, parses and displays the contents
// of the file pointer passed to it
//

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#define BUFSIZE 256

void ParseStringFields(FILE *fp)
{
   char buf[BUFSIZE];

   // place current line into buf
   while (fgets(buf, (int)sizeof(buf), fp))
   {
      // search for delimiters
      for (char *cp = buf; cp = strtok(cp, "AEIOUaeiou\t\n"); cp = NULL)
      {
         int whiteSpaceIdx = 0;

         // find index in cp of first non whitespace character
         while (isspace(cp[whiteSpaceIdx]))
            whiteSpaceIdx++;

         // display line from buffer (excluding leading whitespace)
         printf("%s\n", cp + whiteSpaceIdx);
      }

      ~~// break loop if End of File reached~~
      ~~if (feof(fp))~~
         ~~break;~~
   }
}
```

A "for" statement (note 3.5) is typically the most appropriate choice whenever an object/variable meets at least two of the following criteria:

1. It can be initialized in the loop's "initial expression";
2. It is tested before entering the body of the loop;
3. It is updated after each execution of the loop body.

Unnecessary code

```
********** C2 ASSIGNMENT 3 EXERCISE 4 AUTOMATIC PROGRAM RUN RESULTS **********

************* THE RESULTS BELOW HAVE BEEN PARTIALLY CHECKED AND **************
*************   NO ERRORS WERE FOUND.  HOWEVER, THIS DOES NOT   **************
*************  NECESSARILY MEAN THAT THERE ARE NO ERRORS.  THE  **************
*************  INSTRUCTOR WILL DO A MORE THOROUGH CHECK DURING  **************
*************                 MANUAL GRADING.                   **************


----------------------------- PURPOSE OF 1ST RUN -----------------------------
Verify character parsing display.
-------------------------- CODE CHANGES FOR 1ST RUN --------------------------
Using input file "TestFile10.txt"
------------------------------ START OF 1ST RUN ------------------------------

R
C_N
; ++
)
/* l

p thr
gh
nn

d
d r
c
rds L
P */
f (fsc?
nf(fp, "%*[^\n]%*c") ==
F)
{ /* s

d & thr
w
w
y */



------------------------------- END OF 1ST RUN -------------------------------

----------------------------- PURPOSE OF 2ND RUN -----------------------------
Verify character parsing display.
-------------------------- CODE CHANGES FOR 2ND RUN --------------------------
Using input file "TestFile12.txt"
------------------------------ START OF 2ND RUN ------------------------------


R  / sc
nf[.]/ pr
ntf

/;}? c
n ?.!c
t  ?,;:=+ + C++

n
```

```
,#! t
xt f
l
!/: by
:/\ \



------------------------------- END OF 2ND RUN -------------------------------

----------------------------- PURPOSE OF 3RD RUN -----------------------------
Verify that program detects an input file open failure.
-------------------------- CODE CHANGES FOR 3RD RUN --------------------------
Using input file "bad//file//a"
------------------------------ START OF 3RD RUN ------------------------------

bad//file//a  failed to open



------------------------------- END OF 3RD RUN -------------------------------

----------------------------- PURPOSE OF 4TH RUN -----------------------------
Verify that program detects an input file open failure.
-------------------------- CODE CHANGES FOR 4TH RUN --------------------------
Using input file "bad//file//b"
------------------------------ START OF 4TH RUN ------------------------------

bad//file//b  failed to open



------------------------------- END OF 4TH RUN -------------------------------
```