

Consolidated Assignment 1 Report

This report contains the graded results for the newest of each exercise submitted to the assignment checker prior to 7/15/2020 9:10:07 PM PDT.

Student Name: Shaun Chemplavil
Student ID: U08713628
Contact e-mail: schemp98@hotmail.com
C/C++ Programming II (Section 149123)

Title blocks provide no meaningful information about the purpose/functionality of what is in the files.

Submitted:

Exercise 0: 7/9/2020 4:05:28 PM PDT
Exercise 1: 7/9/2020 4:15:21 PM PDT
Exercise 2: 7/9/2020 4:41:07 PM PDT
Exercise 3: 7/10/2020 4:42:33 PM PDT
Exercise 4: 7/10/2020 5:26:45 PM PDT
Exercise 5: 7/12/2020 10:19:26 AM PDT
Exercise 6: 7/12/2020 9:22:17 AM PDT
Exercise 7: 7/12/2020 12:23:08 PM PDT

-1.5s

Score (out of 20 possible): 17.3

For each exercise in assignments 1 and 2 only, point deductions followed by the letter "s" will be refunded if all assignment checker issues, including runtime issues, are resolved for that exercise and you get a "clean" report before the next assignment is due. If you decide to take advantage of this offer please notify me when you are completely finished with all resubmissions of all exercises you intend to submit - PLEASE DO NOT NOTIFY ME ONE EXERCISE AT A TIME IF YOU ARE CORRECTING MULTIPLE EXERCISES.

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: C2A1E0_U08713628
Submitted: 7/9/2020 4:05:28 PM PDT
Course: C/C++ Programming II (Section 149123)
Student's name: Shaun Chemplavil
Contact email: schemp98@hotmail.com
Student ID: U08713628
Assignment 1, Exercise 0 (C2_0058903M02005X39008)
Exercise point value: 4
File submitted:
C2A1E0_Quiz.txt

NOTE: The assignment checker does not check the correctness of quiz answers for this assignment.

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.

-1.2

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.

07/08/2020

C2A1E0_Quiz.txt

Answers to Quiz 1

1) E <---B

2) D

3) C

4) C <---D

5) C

6) B

7) D

8) D <---A

9) B

10) A

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: C2A1E1_U08713628
Submitted: 7/9/2020 4:15:21 PM PDT
Course: C/C++ Programming II (Section 149123)
Student's name: Shaun Chemplavil
Contact email: schemp98@hotmail.com
Student ID: U08713628
Assignment 1, Exercise 1 (C2_001337124M02005X89337)
Exercise point value: 2
Files submitted:
 C2A1E1_main-Driver.c
 C2A1E1_Macros.h

"Compile-time" results:

No "compile-time" issues;

"Run-time" results:

Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);

```
1 //
2 // Shaun Chemplavil U08713628
3 // shaun.chemplavil@gmail.com
4 // C / C++ Programming II : Dynamic Memory and File I / O Concepts
5 // 149123 Raymond L.Mitchell, Jr., M.S.
6 // 07 / 09 / 2020
7 // C2A1E1_Macros.h
8 // Win10
9 // Visual C++ 19.0
10 //
11 // File containing macros for Exercise 1 of Assignment 1
12 //
13
14 #ifndef C2A1E1_MACROS_H
15 #define C2A1E1_MACROS_H
16
17 // Macro that returns the product of 2 Inputs
18 #define Product(in1, in2) ((in1) * (in2))
19 // Macro that negates the input value
20 #define Negate(in) (-(in))
21
22 // Macro that returns the product of 2 Inputs
23 #define Elements(in) (sizeof(in)/sizeof((in)[0]))
24
25 #endif
```

Title block provides no meaningful information about the purpose/functionality of what is in this file.

***** C2 ASSIGNMENT 1 EXERCISE 1 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 -----

C2A1E1: 6 tests 0 errors.

----- 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: C2A1E2_U08713628
Submitted: 7/9/2020 4:41:07 PM PDT
Course: C/C++ Programming II (Section 149123)
Student's name: Shaun Chemplavil
Contact email: schemp98@hotmail.com
Student ID: U08713628
Assignment 1, Exercise 2 (C2_001603483M02005X3603)
Exercise point value: 2
File submitted:
C2A1E2_main.c

"Compile-time" results:

No "compile-time" issues;

"Run-time" results:

Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);

```
1  //
2  // Shaun Chemplavil U08713628
3  // shaun.chemplavil@gmail.com
4  // C / C++ Programming II : Dynamic Memory and File I / O Concepts
5  // 149123 Raymond L.Mitchell, Jr., M.S.
6  // 07 / 09 / 2020
7  // C2A1E2_main.c
8  // Win10
9  // Visual C++ 19.0
10 //
11 // File containing Exercise 2 of Assignment 1
12 //
13
14 #include <stdio.h>
15
16 int main(int argc, char *argv[])
17 {
18     // Display the number of input arguments
19     printf("%d\n", argc);
20
21     // Print each input argument
22     for (int arg_num = 0; arg_num < argc; arg_num++)
23     {
24         printf("%s\n", argv[arg_num]);
25     }
26
27     return 0;
28 }
```

Title block provides no meaningful information about the purpose/functionality of what is in this file.

***** C2 ASSIGNMENT 1 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. *****
```

---- ADDITIONAL COMMAND LINE ARGUMENTS FOR 1ST RUN STARTING WITH argv[1] ----
<no additional arguments>

----- START OF 1ST RUN -----

1
S:\AutoGrade__RunnableExec__AutoGrade_637299098124.exe

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

---- ADDITIONAL COMMAND LINE ARGUMENTS FOR 2ND RUN STARTING WITH argv[1] ----
argv1 argv2 argv3

----- START OF 2ND RUN -----

4
S:\AutoGrade__RunnableExec__AutoGrade_637299098144.exe
argv1
argv2
argv3

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

---- ADDITIONAL COMMAND LINE ARGUMENTS FOR 3RD RUN STARTING WITH argv[1] ----
"A B C D" "E F G H" "I J K L" "M N O P" "Q R S T"

----- START OF 3RD RUN -----

6
S:\AutoGrade__RunnableExec__AutoGrade_637299098164.exe
A B C D
E F G H
I J K L
M N O P
Q R S T

----- 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: C2A1E3_U08713628
Submitted: 7/10/2020 4:42:33 PM PDT
Course: C/C++ Programming II (Section 149123)
Student's name: Shaun Chemplavil
Contact email: schemp98@hotmail.com
Student ID: U08713628
Assignment 1, Exercise 3 (C2_001117507M02005X68117)
Exercise point value: 2
Files submitted:
 C2A1E3_main-Driver.c
 C2A1E3_FindFirstInt.c

"Compile-time" results:

No "compile-time" issues;

"Run-time" results:

Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);

```
1  //
2  // Shaun Chemplavil U08713628
3  // shaun.chemplavil@gmail.com
4  // C / C++ Programming II : Dynamic Memory and File I / O Concepts
5  // 149123 Raymond L.Mitchell, Jr., M.S.
6  // 07 / 10 / 2020
7  // C2A1E3_FindFirstInt.c
8  // Win10
9  // Visual C++ 19.0
10 //
11 // File containing FindFirstInt function
12 //
13
14 #include <stdio.h>
15
16 int *FindFirstInt(const int *ptr, size_t count, int value)
17 {
18     // cycle through array values
19     for (int index = 0; index < (int)count; index++)
20     {
21         // When desired value encountered, return pointer
22         if (ptr[index] == value)
23             return((int *)&ptr[index]);
24     }
25
26     // If value is not found return NULL
27     return(NULL);
28 }
```

Title block provides no meaningful information about the purpose/functionality of what is in this file.

***** C2 ASSIGNMENT 1 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. *****
```

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

C2A1E3: 2 tests 0 errors.

----- 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: C2A1E4_U08713628
Submitted: 7/10/2020 5:26:45 PM PDT
Course: C/C++ Programming II (Section 149123)
Student's name: Shaun Chemplavil
Contact email: schemp98@hotmail.com
Student ID: U08713628
Assignment 1, Exercise 4 (C2_001134932M02005X23134)
Exercise point value: 2
Files submitted:
 C2A1E4_main-Driver.c
 C2A1E4_StrToUpper.c

"Compile-time" results:

No "compile-time" issues;

"Run-time" results:

Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);

```

1  //
2  // Shaun Chemplavil U08713628
3  // shaun.chemplavil@gmail.com
4  // C / C++ Programming II : Dynamic Memory and File I / O Concepts
5  // 149123 Raymond L.Mitchell, Jr., M.S.
6  // 07 / 10 / 2020
7  // C2A1E4_StrToUpper.c
8  // Win10
9  // Visual C++ 19.0
10 //
11 // File containing StrToUpper function
12 //
13
14 #include <ctype.h>
15
16 size_t StrToUpper(char destination[], const char source[])
17 {
18     // Keep track of initial pointer location
19     const char * const START = source;
20
21     // Increment source pointer until the null character is reached
22     while (*source)
23         *destination++ = (char)toupper(*source++);
24
25     // Append Null Character
26     *destination = *source;
27
28     // Determine the total length of input string
29     return (size_t)(source - START);
30 }

```

Title block provides no meaningful information about the purpose/functionality of what is in this file.

***** C2 ASSIGNMENT 1 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. *****
```

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

C2A1E4: 4 tests 0 errors.

----- 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: C2A1E5_U08713628
Submitted: 7/12/2020 10:19:26 AM PDT
Course: C/C++ Programming II (Section 149123)
Student's name: Shaun Chemplavil
Contact email: schemp98@hotmail.com
Student ID: U08713628
Assignment 1, Exercise 5 (C2_001741980M02005X70741)
Exercise point value: 2
Files submitted:
 C2A1E5_main-Driver.c
 C2A1E5_ResizeAlloc.c

"Compile-time" results:

No "compile-time" issues;

"Run-time" results:

Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);


```
1  //
2  // Shaun Chemplavil U08713628
3  // shaun.chemplavil@gmail.com
4  // C / C++ Programming II : Dynamic Memory and File I / O Concepts
5  // 149123 Raymond L. Mitchell, Jr., M.S.
6  // 07 / 12 / 2020
7  // C2A1E5_ResizeAlloc.c
8  // Win10
9  // Visual C++ 19.0
10 //
11 // File containing ResizeAlloc function
12 //
13
14 #include <stdlib.h>
15 #include <string.h>
16 #include <stdio.h>
17
18 void *ResizeAlloc(void *pOld, size_t newSize, size_t oldSize)
19 {
20     if (newSize == 0)
21         return(NULL);
22     else
23     {
24         void *pNew;
25         // allocate newSize bytes of memory to pNew, test for success
26         if ((pNew = malloc(newSize)) == NULL)
27         {
28             fputs("Not enough memory for pNew\n", stderr);
29             exit(EXIT_FAILURE);
30         }
31
32         // Exit if no data at pOld memory location
33         if (pOld == NULL)
34             return(pNew);
35
36         // Keep track number of bytes to copy into newly allocated memory
37         size_t copySize;
38
39         // Ensure that we do not copy more bytes that we need to while not
40         // exceeding the allocated memory
41         if (newSize > oldSize)
42             copySize = oldSize;
43         else
44             copySize = newSize;
45
46         memcpy(pNew, pOld, copySize);
47
48         free(pOld);
49
50         return(pNew);
51     }
52 }
```

Title block provides no meaningful information about the purpose/functionality of what is in this file.

***** C2 ASSIGNMENT 1 EXERCISE 5 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 that allocation was resized.
----- START OF 1ST RUN -----
```

C2A1E5: 7 tests 0 errors.

```
----- END OF 1ST 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: C2A1E6_U08713628
Submitted: 7/12/2020 9:22:17 AM PDT
Course: C/C++ Programming II (Section 149123)
Student's name: Shaun Chemplavil
Contact email: schemp98@hotmail.com
Student ID: U08713628
Assignment 1, Exercise 6 (C2_001356141M02005X59356)
Exercise point value: 2
Files submitted:
 C2A1E6_AppendFile.cpp
 C2A1E6_main-Driver.cpp

"Compile-time" results:

No "compile-time" issues;

"Run-time" results:

Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);

```
1  //
2  // Shaun Chemplavil U08713628
3  // shaun.chemplavil@gmail.com
4  // C / C++ Programming II : Dynamic Memory and File I / O Concepts
5  // 149123 Raymond L.Mitchell, Jr., M.S.
6  // 07 / 12 / 2020
7  // C2A1E6_AppendFile.cpp
8  // Win10
9  // Visual C++ 19.0
10 //
11 // File containing definition of the AppendFile function
12 //
13
14 #include <iostream>
15 #include <fstream>
16
17 using namespace std;
18
19 const int BUFSIZE = 256;
20
21 static void ErrorAndQuit(const char *myString)
22 {
23     cerr << "\"" << myString << "\" :File access error!\n";
24     exit(-1);
25 }
26
27 int AppendFile(const char *inFile, const char *outFile)
28 {
29     // Keep track of the bytes read from binary files
30     streamsize bytesRead;
31
32     // open inFile in "read" and binary mode
33     ifstream source(inFile, ios_base::binary);
34     if (!source.is_open())
35         ErrorAndQuit(inFile);
36
37     // open outFile in "append" and binary mode
38     ofstream destination(outFile, ios_base::app | ios_base::binary);
39     if (!destination.is_open())
40     {
41         // close all open files because an error has occurred
42         source.close();
43         ErrorAndQuit(outFile);
44     }
45
46     do
47     {
48         // Buffer to hold file contents
49         char buf[BUFSIZE];
50
51         // Read block of data from source into buffer
52         source.read(buf, sizeof(buf));
53
54         if (bytesRead = source.gcount())
55             // write buffer into destination file
56             destination.write(buf, bytesRead);
57     } while (bytesRead == BUFSIZE);
58
59     // Close open files
60     source.close();
61     destination.close();
```

Title block provides no meaningful information about the purpose/functionality of what is in this file.

```
62  
63  
64  
1  
return(0);  
}
```

***** C2 ASSIGNMENT 1 EXERCISE 6 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 -----

C2A1E6: 2 tests 0 errors.

----- 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: C2A1E7_U08713628
Submitted: 7/12/2020 12:23:08 PM PDT
Course: C/C++ Programming II (Section 149123)
Student's name: Shaun Chemplavil
Contact email: schemp98@hotmail.com
Student ID: U08713628
Assignment 1, Exercise 7 (C2_001136030M02005X85136)
Exercise point value: 4
Files submitted:
 C2A1E7_main-Driver.cpp
 C2A1E7_Employee.h
 C2A1E7_Employee.cpp

"Compile-time" results:

No "compile-time" issues;

"Run-time" results:

Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);

```
1 //
2 // Shaun Chemplavil U08713628
3 // shaun.chemplavil@gmail.com
4 // C / C++ Programming II : Dynamic Memory and File I / O Concepts
5 // 149123 Raymond L.Mitchell, Jr., M.S.
6 // 07 / 11 / 2020
7 // C2A1E7_Employee.h
8 // Win10
9 // Visual C++ 19.0
10 //
11 // Header File Employee Class definition and Get Member Function
12 //
13
14 #ifndef C2A1E7_EMPLOYEE_H
15 #define C2A1E7_EMPLOYEE_H
16
17 class Employee
18 {
19     char *name;
20     int    age;
21     float  raise;
22     double salary;
23
24 public:
25
26     // Definition of this prototype in C2A1E7_Employee.cpp
27     // used to set member variable name
28     void Set(const char *input);
29
30     // Set Member Variable raise
31     void Set(int input = 25)
32     {
33         age = input;
34     }
35
36     // Set Member Variable raise
37     void Set(const float &input)
38     {
39         raise = input;
40     }
41
42     // Set Member Variable salary
43     void Set(const double *input)
44     {
45         salary = *input;
46     }
47
48     // Constant Member Functions //
49
50     // Get Member Variable name
51     char *Get(char **input) const
52     {
53         *input = name;
54
55         return name;
56     }
57
58     // Get Member Variable age
59     int Get(int &input) const
60     {
61         input = age;
```



```
1
62
63     return age;
64 }
65 // Get Member Variable raise
66 float &Get(float &input)
67 {
68     input = raise;
69
70     return raise;
71 }
72
73 // Definition of this member function prototype below
74 inline double Get(double *input) const;
75 };
76
77 // Member function to get salary information
78 double Employee::Get(double *input) const
79 {
80     *input = salary;
81     return salary;
82 }
83
84 #endif
```

```

1  //
2  // Shaun Chemplavil U08713628
3  // shaun.chemplavil@gmail.com
4  // C / C++ Programming II : Dynamic Memory and File I / O Concepts
5  // 149123 Raymond L.Mitchell, Jr., M.S.
6  // 07 / 11 / 2020
7  // C2A1E7_Employee.cpp
8  // Win10
9  // Visual C++ 19.0
10 //
11 // File Contains an overloaded definition of Employee:Set member function
12 //
13
14 #include "C2A1E7_Employee.h"
15 #include <iostream>
16 #include <fstream>
17 #include <string>
18
19 using namespace std;
20
21 void Employee::Set(const char *input)
22 {
23     // When determining the size of the character string need to account for
24     // null character by adding one to the output of strlen
25     size_t numChar = strlen(input) + 1;
26     char *temp;
27
28     // Allocate memory for temp variable
29     temp = new char[numChar];
30
31     // verify successful allocation
32     if (temp == nullptr)
33     {
34         cerr << "Failed to allocate memory for temp!\n";
35         exit(EXIT_FAILURE);
36     }
37
38     // point name to temp
39     name = temp;
40
41     // copy data located at input pointer into newly allocated memory
42     memcpy(temp, input, numChar);
43 }

```

Title block provides no meaningful information about the purpose/functionality of what is in this file.

See note 8.5.

That's not what this does.

***** C2 ASSIGNMENT 1 EXERCISE 7 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 -----

C2A1E7: 9 tests 0 errors.

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