Consolidated Assignment 5 Report

This report contains the graded results for the newest of each exercise submitted to the assignment checker prior to 5/6/2020 3:23:44 AM PDT.

Student Name: Shaun Chemplavil
Student ID: U08713628
Contact e-mail: shaun.chemplavil@gmail.com
C/C++ Programming I (Section 146359)

Erroneous/misleading comments: No comments at all are preferable to erroneous or misleading comments.

-0.5

Submitted:
   Exercise 0: 4/18/2020 7:51:25 AM PDT
   Exercise 1: 4/18/2020 10:32:05 AM PDT
   Exercise 2: 5/1/2020 7:22:35 AM PDT
   Exercise 3: 5/1/2020 7:22:52 AM PDT

Score (out of 20 possible): 18.5

```
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
"Announcements" page of the course 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: C1A5E0_U08713628
   Submitted: 4/18/2020 7:51:25 AM PDT
   Course: C/C++ Programming I (Section 146359)
   Student's name: Shaun Chemplavil
   Contact email: shaun.chemplavil@gmail.com
   Student ID: U08713628
   Assignment 5, Exercise 0
   Exercise point value: 6
   File submitted:
      C1A5E0_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

```
Shaun Chemplavil U08713628
shaun.chemplavil@gmail.com
C/C++ Programming I : Fundamental Programming Concepts
146359 Raymond L. Mitchell, Jr., M.S.
04/18/2020
C1A5E0_Quiz.txt
Answers to Quiz

1. C
2. B
3. D <---B
4. E
5. D
6. C
```

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 "Announcements" page of the course 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: C1A5E1_U08713628
   Submitted: 4/18/2020 10:32:05 AM PDT
   Course: C/C++ Programming I (Section 146359)
   Student's name: Shaun Chemplavil
   Contact email: shaun.chemplavil@gmail.com
   Student ID: U08713628
   Assignment 5, Exercise 1
   Exercise point value: 6
   File submitted:
      C1A5E1_main.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 I : Fundamental Programming Concepts
// 146359 Raymond L. Mitchell Jr.
// 04 / 18 / 2020
// C1A5E1_main.c
// Win10
// Visual C++ 19.0
//
// This file will implement a survey to customers to indicate how much they
// enjoy a product, and then display results of all participants
//

#include <stdio.h>

#define MAX_RESPONDENTS      3                    -
#define MIN_RESPONSE_VALUE  (-3)
#define MAX_RESPONSE_VALUE  (-3)
#define OUT_OF_RANGE_LIMIT   3
#define RESPONSE_VALUES (1 + MAX_RESPONSE_VALUE + MIN_RESPONSE_VALUE)

int main(void)
{
   int consecutiveRangeErrors = 0, ratingDisp = MAX_RESPONSE_VALUE;
   int ratingCounters[RESPONSE_VALUES] = {0};

   // Request and Store User Input
   for (int numResponses = 0; numResponses < MAX_RESPONDENTS;)
   {
      int rating;

      // get user rating
      printf("Enter your rating: ");
      scanf("%d", &rating);

      // check rating validity... if invalid, prompt new response a maximum
      // of OUT_OF_RANGE_LIMIT times
      if (rating >= MIN_RESPONSE_VALUE && rating <= MAX_RESPONSE_VALUE)
      {
         consecutiveRangeErrors = 0;
         ++ratingCounters[MAX_RESPONSE_VALUE - rating];
         ++numResponses;
      }
      else // Invalid response
      {
         ++consecutiveRangeErrors;

         if (consecutiveRangeErrors < OUT_OF_RANGE_LIMIT)
         {
            printf("Rating out of range, Try again!\n");
            continue;
         }
         else
         {
            printf("Reached MAX Out of Range entries, ending survey\n");
            break;
         }
      }
   }

```

This statement serves no purpose. What was your reason for using it?

```
   // Display summary of ratings
   printf(
      "Rating  Responses\n"
      "------  ----------\n");
   for (int rating = 0; rating < RESPONSE_VALUES; rating++)
   {
      printf("%5d    %4d\n", ratingDisp, ratingCounters[rating]);
      --ratingDisp;
   }

   return 0;
}
```

```
********** C1 ASSIGNMENT 5 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  **************
*************   THE INSTRUCTOR WILL DO A MORE THOROUGH CHECK    **************
*************              DURING MANUAL GRADING.               **************


-------------------------- CODE CHANGES FOR 1ST RUN --------------------------
MAX_RESPONDENTS = 3  OUT_OF_RANGE_LIMIT = 3
MIN_RESPONSE_VALUE = -3  MAX_RESPONSE_VALUE = -3
------------------------------ START OF 1ST RUN ------------------------------

Enter your rating: -3
Enter your rating: -4
Rating out of range, Try again!
Enter your rating: -3
Enter your rating: -2
Rating out of range, Try again!
Enter your rating: 0
Rating out of range, Try again!
Enter your rating: -3
Rating  Responses
------  ----------
   -3       3

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

-------------------------- CODE CHANGES FOR 2ND RUN --------------------------
MAX_RESPONDENTS = 10  OUT_OF_RANGE_LIMIT = 2
MIN_RESPONSE_VALUE = 0  MAX_RESPONSE_VALUE = 3
------------------------------ START OF 2ND RUN ------------------------------

Enter your rating: 0
Enter your rating: 1
Enter your rating: 1
Enter your rating: 2
Enter your rating: -1
Rating out of range, Try again!
Enter your rating: 2
Enter your rating: 2
Enter your rating: -1
Rating out of range, Try again!
Enter your rating: -2
Reached MAX Out of Range entries, ending survey
Rating  Responses
------  ----------
    3       0
    2       3
    1       2
    0       1

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

-------------------------- CODE CHANGES FOR 3RD RUN --------------------------
MAX_RESPONDENTS = 1  OUT_OF_RANGE_LIMIT = 1
MIN_RESPONSE_VALUE = 1  MAX_RESPONSE_VALUE = 1
------------------------------ START OF 3RD RUN ------------------------------
```

```
Enter your rating: 1
Rating  Responses
------  ----------
    1       1

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

-------------------------- CODE CHANGES FOR 4TH RUN --------------------------
MAX_RESPONDENTS = 1  OUT_OF_RANGE_LIMIT = 1
MIN_RESPONSE_VALUE = 1  MAX_RESPONSE_VALUE = 1
------------------------------ START OF 4TH RUN ------------------------------

Enter your rating: 0
Reached MAX Out of Range entries, ending survey
Rating  Responses
------  ----------
    1       0

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

-------------------------- CODE CHANGES FOR 5TH RUN --------------------------
MAX_RESPONDENTS = 5  OUT_OF_RANGE_LIMIT = 4
MIN_RESPONSE_VALUE = -1  MAX_RESPONSE_VALUE = 2
------------------------------ START OF 5TH RUN ------------------------------

Enter your rating: -1
Enter your rating: 0
Enter your rating: 0
Enter your rating: -27
Rating out of range, Try again!
Enter your rating: -27
Rating out of range, Try again!
Enter your rating: -27
Rating out of range, Try again!
Enter your rating: -27
Reached MAX Out of Range entries, ending survey
Rating  Responses
------  ----------
    2       0
    1       0
    0       2
   -1       1

------------------------------- END OF 5TH RUN -------------------------------

-------------------------- CODE CHANGES FOR 6TH RUN --------------------------
MAX_RESPONDENTS = 13  OUT_OF_RANGE_LIMIT = 3
MIN_RESPONSE_VALUE = -10  MAX_RESPONSE_VALUE = -4
------------------------------ START OF 6TH RUN ------------------------------

Enter your rating: -10
Enter your rating: -4
Enter your rating: -10
Enter your rating: -4
Enter your rating: -10
Enter your rating: -4
Enter your rating: -4
Enter your rating: -4
Enter your rating: -4
Enter your rating: -4
Enter your rating: -4
Enter your rating: -4
Enter your rating: -4
Rating  Responses
------  ----------
   -4      10
   -5       0
   -6       0
   -7       0
   -8       0
   -9       0
  -10       3

------------------------------- END OF 6TH 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
"Announcements" page of the course 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: C1A5E2_U08713628
   Submitted: 5/1/2020 7:22:35 AM PDT
   Course: C/C++ Programming I (Section 146359)
   Student's name: Shaun Chemplavil
   Contact email: shaun.chemplavil@gmail.com
   Student ID: U08713628
   Assignment 5, Exercise 2
   Exercise point value: 4
   Files submitted:
      C1A5E2_main.cpp
      C1A5E2_ComputeMaximum.cpp
      C1A5E2_ComputeMinimum.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 I : Fundamental Programming Concepts
// 146359 Raymond L. Mitchell Jr.
// 05 / 01 / 2020
// C1A5E2_ComputeMaximum.cpp
// Win10
// Visual C++ 19.0
//
// This function returns the maximum of two type double arguments
//

double &ComputeMaximum(const double &input1, const double &input2)
{
   return (double &)((input1 > input2) ? input1 : input2);
}
```

Erroneous/misleading comment: No comments at all are preferable to erroneous or misleading comments.

```
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C / C++ Programming I : Fundamental Programming Concepts
// 146359 Raymond L. Mitchell Jr.
// 05 / 01 / 2020
// C1A5E2_ComputeMinimum.cpp
// Win10
// Visual C++ 19.0
//
// This function returns the minimum of two type double arguments
//

double &ComputeMinimum(const double &input1, const double &input2)
{
   return (double &)((input1 < input2) ? input1 : input2);
}
```

Erroneous/misleading comment: No comments at all are preferable to erroneous or misleading comments.

```cpp
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C / C++ Programming I : Fundamental Programming Concepts
// 146359 Raymond L. Mitchell Jr.
// 05 / 01 / 2020
// C1A5E2_main.cpp
// Win10
// Visual C++ 19.0
//
// This file will take two decimal numbers and display their
//  maximum and minimum values by accessing their memory addresses
//

#include <iostream>

using namespace std;

double &ComputeMinimum(const double &input1, const double &input2);
double &ComputeMaximum(const double &input1, const double &input2);

int main()
{
   double input1, input2;

   // Request and Store User Input
   cout << "Enter two space-separated decimal values on the same line: ";
   cin >> input1 >> input2;

   // Display Minimum and Maximum values
   cout <<
      "ComputeMinimum(" << input1 << "," << input2 << ") returned " <<
      ComputeMinimum(input1, input2) << "\n"
      "ComputeMaximum(" << input1 << "," << input2 << ") returned " <<
      ComputeMaximum(input1, input2) << "\n";

   return 0;
}
```

```
********** C1 ASSIGNMENT 5 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 1ST RUN ------------------------------

Enter two space-separated decimal values on the same line: 3 3
ComputeMinimum(3,3) returned 3
ComputeMaximum(3,3) returned 3


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

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

Enter two space-separated decimal values on the same line: -7.98 7.13
ComputeMinimum(-7.98,7.13) returned -7.98
ComputeMaximum(-7.98,7.13) returned 7.13


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

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

Enter two space-separated decimal values on the same line: 2000.45 0
ComputeMinimum(2000.45,0) returned 0
ComputeMaximum(2000.45,0) returned 2000.45


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

------------------------------ START OF 4TH RUN ------------------------------

Enter two space-separated decimal values on the same line: 54e-2 86e-1
ComputeMinimum(0.54,8.6) returned 0.54
ComputeMaximum(0.54,8.6) returned 8.6


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

------------------------------ START OF 5TH RUN ------------------------------

Enter two space-separated decimal values on the same line: 86e-1 54e-2
ComputeMinimum(8.6,0.54) returned 0.54
ComputeMaximum(8.6,0.54) returned 8.6


------------------------------- END OF 5TH RUN -------------------------------

------------------------------ START OF 6TH RUN ------------------------------

Enter two space-separated decimal values on the same line: -0 0
ComputeMinimum(-0,0) returned 0
ComputeMaximum(-0,0) returned 0
```

```
------------------------------- END OF 6TH 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 "Announcements" page of the course 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: C1A5E3_U08713628
   Submitted: 5/1/2020 7:22:52 AM PDT
   Course: C/C++ Programming I (Section 146359)
   Student's name: Shaun Chemplavil
   Contact email: shaun.chemplavil@gmail.com
   Student ID: U08713628
   Assignment 5, Exercise 3
   Exercise point value: 4
   Files submitted:
      C1A5E3_ComputeMaximum.cpp
      C1A5E3_ComputeMinimum.cpp
      C1A5E3_main.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 I : Fundamental Programming Concepts
// 146359 Raymond L. Mitchell Jr.
// 05 / 01 / 2020
// C1A5E3_ComputeMaximum.cpp
// Win10
// Visual C++ 19.0
//
// This function returns the maximum of two type double arguments
//

double *ComputeMaximum(const double *input1, const double *input2)
{
   return (double *)((*input1 > *input2) ? input1 : input2);
}
```

Erroneous/misleading comment: No comments at all are preferable to erroneous or misleading comments.

```
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C / C++ Programming I : Fundamental Programming Concepts
// 146359 Raymond L. Mitchell Jr.
// 05 / 01 / 2020
// C1A5E3_ComputeMinimum.cpp
// Win10
// Visual C++ 19.0
//
// This function returns the minimum of two type double arguments
//

double *ComputeMinimum(const double *input1, const double *input2)
{
    return (double *)((*input1 < *input2) ? input1 : input2);
}
```

Erroneous/misleading comment: No comments at all are preferable to erroneous or misleading comments.

```cpp
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C / C++ Programming I : Fundamental Programming Concepts
// 146359 Raymond L. Mitchell Jr.
// 05 / 01 / 2020
// C1A5E3_main.cpp
// Win10
// Visual C++ 19.0
//
// This file will take two decimal numbers and display their
//  maximum and minimum values via pointers to their values
//

#include <iostream>

using namespace std;

double *ComputeMinimum(const double *input1, const double *input2);
double *ComputeMaximum(const double *input1, const double *input2);

int main()
{
   double input1, input2;

   // Request and Store User Input
   cout << "Enter two space-separated decimal values on the same line: ";
   cin >> input1 >> input2;

   // Display Minimum and Maximum values
   cout <<
      "ComputeMinimum(&" << input1 << ",&" << input2 << ") returned &" <<
      *ComputeMinimum(&input1, &input2) << "\n"
      "ComputeMaximum(&" << input1 << ",&" << input2 << ") returned &" <<
      *ComputeMaximum(&input1, &input2) << "\n";

   return 0;
}
```

```
********** C1 ASSIGNMENT 5 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 1ST RUN ------------------------------

Enter two space-separated decimal values on the same line: 3 3
ComputeMinimum(&3,&3) returned &3
ComputeMaximum(&3,&3) returned &3


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

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

Enter two space-separated decimal values on the same line: -7.98 7.13
ComputeMinimum(&-7.98,&7.13) returned &-7.98
ComputeMaximum(&-7.98,&7.13) returned &7.13


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

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

Enter two space-separated decimal values on the same line: 2000.45 0
ComputeMinimum(&2000.45,&0) returned &0
ComputeMaximum(&2000.45,&0) returned &2000.45


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

------------------------------ START OF 4TH RUN ------------------------------

Enter two space-separated decimal values on the same line: 54e-2 86e-1
ComputeMinimum(&0.54,&8.6) returned &0.54
ComputeMaximum(&0.54,&8.6) returned &8.6


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

------------------------------ START OF 5TH RUN ------------------------------

Enter two space-separated decimal values on the same line: 86e-1 54e-2
ComputeMinimum(&8.6,&0.54) returned &0.54
ComputeMaximum(&8.6,&0.54) returned &8.6


------------------------------- END OF 5TH RUN -------------------------------

------------------------------ START OF 6TH RUN ------------------------------

Enter two space-separated decimal values on the same line: -0 0
ComputeMinimum(&-0,&0) returned &0
ComputeMaximum(&-0,&0) returned &0
```

```
------------------------------- END OF 6TH RUN -------------------------------
```