Consolidated Assignment 2 Report

This report contains the graded results for the newest of each exercise submitted to the assignment checker prior to 4/15/2020 1:08:59 PM PDT.

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

Submitted:
- Exercise 0: 4/3/2020 8:37:39 AM PDT
- Exercise 1: 4/3/2020 4:44:58 PM PDT
- Exercise 2: 4/10/2020 1:52:39 PM PDT
- Exercise 3: 4/10/2020 1:57:17 PM PDT

Score (out of 20 possible): 19

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:schemp98@hotmail.com>
Subject: C1A2E0_U08713628
Submitted: 4/3/2020 8:37:39 AM PDT
Course: C/C++ Programming I (Section 146359)
Student's name: Shaun Chemplavil
Contact email: schemp98@hotmail.com
Student ID: U08713628
Assignment 2, Exercise 0
Exercise point value: 6
File submitted:
C1A2E0_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
schemp98@hotmail.com
C/C++ Programming I : Fundamental Programming Concepts
146359 Raymond L. Mitchell, Jr., M.S.
04/03/2020
C1A2E0_Quiz.txt
Answers to Quiz 2

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

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:schemp98@hotmail.com>
Subject: C1A2E1_U08713628
Submitted: 4/3/2020 4:44:58 PM PDT
Course: C/C++ Programming I (Section 146359)
Student's name: Shaun Chemplavil
Contact email: schemp98@hotmail.com
Student ID: U08713628
Assignment 2, Exercise 1
Exercise point value: 5
File submitted:
C1A2E1_main.cpp

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

```cpp
//
// Shaun Chemplavil U08713628
// schemp98@hotmail.com
// C / C++ Programming I : Fundamental Programming Concepts
// 146359 Raymond L.Mitchell, Jr., M.S.
// 04 / 03 / 2020
// C1A2E1_main.cpp
// Win10
// Visual C++ 19.0
//
// This file converts a user-input character to lowercase
// This is accomplished by adding "TO_LOWER" to the integer value of the input
// ("TO_LOWER" = '32' is the difference between upper and lower case letters
//   Code entries in the ASCII table)
// Off-nominal Case 1: If a non-Uppercase character is entered, the character
//                     32 codes higher than the entered character will be output
// Off-nominal Case 2: If a white-space character is entered, the character
//                     32 codes higher than the FIRST whitespace character
//                     entered will be output
//

#include <iostream>
using namespace std;

int main()
{
   // Set variable to convert from upper to lowercase
   const int TO_LOWER = 'a' - 'A';
   char ch;

   cout << "Enter any character: ";

   // Grab character from user
   cin.get(ch);

   // Convert to lowercase (by adding ASCII Code offset) and output to screen
   cout
      << "\nThe lowercase equivalent of '" << ch << "' is '"
      << char(ch + TO_LOWER) << "'\n";

   return 0;
}
```

```
********** C1 ASSIGNMENT 2 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.                **************


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

Enter any character: A

The lowercase equivalent of 'A' is 'a'

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

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

Enter any character: Z

The lowercase equivalent of 'Z' is 'z'

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

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

Enter any character: 1

The lowercase equivalent of '1' is 'Q'

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

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

Enter any character:

The lowercase equivalent of ' ' is '@'

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

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

Enter any character: ?

The lowercase equivalent of '?' is '_'

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

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

Enter any character: @

The lowercase equivalent of '@' is '`'

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

------------------------------ START OF 7TH RUN ------------------------------

Enter any character: ]
```

```
The lowercase equivalent of ']' is '}'

------------------------------ END OF 7TH 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: <mailto:shaun.chemplavil@gmail.com>
Subject: C1A2E2_U08713628
Submitted: 4/10/2020 1:52:39 PM PDT
Course: C/C++ Programming I (Section 146359)
Student's name: Shaun Chemplavil
Contact email: shaun.chemplavil@gmail.com
Student ID: U08713628
Assignment 2, Exercise 2
Exercise point value: 5
File submitted:
C1A2E2_main.c

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

```
1
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C / C++ Programming I : Fundamental Programming Concepts
// 146359 Raymond L. Mitchell Jr.
// 04 / 10 / 2020
// C1A2E2_main.c
// Win10
// Visual C++ 19.0
//
// This file will take user integer input (input), and display a triangle of
//  characters on the screen
//

#include <stdio.h>

// Set Character to illustrate the "diagonal" of the triangle
#define DIAGONAL_CHAR '\\'
//  Set the Most Significant Bit value
#define RADIX 10

int main(void)
{
   // Declare User Input (input) and Display Character (disp_char)
   int input, disp_char;

   // Request and Store User Input
   printf("Enter any positive integer value: ");
   scanf("%d", &input);

   // Set the first character to be displayed
   disp_char = input;

   // Display triangle
   // loop that controls current row display
   for (int row_idx = 0; row_idx < input; ++row_idx)
   {
      // loop that controls current column display
      for (int col_idx = 0; col_idx < row_idx; ++col_idx)
      {
         // Print Current 'disp_char', increment, and enforce RADIX
         printf("%d", disp_char++ % RADIX);
      }
      // Add DIAGONAL_CHAR
      printf("%c\n", DIAGONAL_CHAR);
   }
   return 0;
}
```

Neither bits nor bytes enter into this exercise algorithm.

What does this mean?

```
********** C1 ASSIGNMENT 2 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  **************
************   THE INSTRUCTOR WILL DO A MORE THOROUGH CHECK    **************
************              DURING MANUAL GRADING.                 **************


------------------------- CODE CHANGES FOR 1ST RUN -------------------------
DIAGONAL_CHAR = '@'
------------------------------ START OF 1ST RUN ------------------------------

Enter any positive integer value: 2
@
2@

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

------------------------- CODE CHANGES FOR 2ND RUN -------------------------
DIAGONAL_CHAR = '@'
------------------------------ START OF 2ND RUN ------------------------------

Enter any positive integer value: 4
@
4@
56@
789@

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

------------------------- CODE CHANGES FOR 3RD RUN -------------------------
DIAGONAL_CHAR = '='
------------------------------ START OF 3RD RUN ------------------------------

Enter any positive integer value: 1
=

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

------------------------- CODE CHANGES FOR 4TH RUN -------------------------
DIAGONAL_CHAR = '.'
------------------------------ START OF 4TH RUN ------------------------------

Enter any positive integer value: 10
.
0.
12.
345.
6789.
01234.
567890.
1234567.
89012345.
678901234.

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

------------------------- CODE CHANGES FOR 5TH RUN -------------------------
DIAGONAL_CHAR = '+'
```

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

Enter any positive integer value: 6
+
6+
78+
901+
2345+
67890+

------------------------------- END OF 5TH 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: <mailto:shaun.chemplavil@gmail.com>
Subject: C1A2E3_U08713628
Submitted: 4/10/2020 1:57:17 PM PDT
Course: C/C++ Programming I (Section 146359)
Student's name: Shaun Chemplavil
Contact email: shaun.chemplavil@gmail.com
Student ID: U08713628
Assignment 2, Exercise 3
Exercise point value: 4
File submitted:
C1A2E3_main.cpp

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

```
1
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C / C++ Programming I : Fundamental Programming Concepts
// 146359 Raymond L.Mitchell, Jr., M.S.
// 04 / 10 / 2020
// C1A2E3_main.cpp
// Win10
// Visual C++ 19.0
//
// This file will take user integer input, and display a triangle of
//  characters on the screen
//

#include <iostream>
using namespace std;

// Set Character to illustrate the "diagonal" of the triangle
const char DIAGONAL_CHAR = '\\';
// Declare and set RADIX;
const int RADIX = 10;

int main()
{
   // Declare User Input (input) and Display Character (disp_char)
   int input, disp_char;

   // Request and Store User Input
   cout << "Enter any positive integer value: ";
   cin >> input;

   // Set the first character to be displayed
   disp_char = input;

   // Display triangle
   // loop that controls current row display
   for (int row_idx = 0; row_idx < input; ++row_idx)
   {
      // loop that controls current column display
      for (int col_idx = 0; col_idx < row_idx; ++col_idx)
      {
         // Print Current 'disp_char', increment, and enforce RADIX
         cout << disp_char++ % RADIX;
      }
      // Add DIAGONAL_CHAR
      cout << DIAGONAL_CHAR << '\n';
   }
   return 0;
}
```

```
********** C1 ASSIGNMENT 2 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  **************
************   THE INSTRUCTOR WILL DO A MORE THOROUGH CHECK    **************
************              DURING MANUAL GRADING.                 **************


------------------------- CODE CHANGES FOR 1ST RUN -------------------------
DIAGONAL_CHAR = '@'
------------------------------ START OF 1ST RUN ------------------------------

Enter any positive integer value: 2
@
2@

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

------------------------- CODE CHANGES FOR 2ND RUN -------------------------
DIAGONAL_CHAR = '@'
------------------------------ START OF 2ND RUN ------------------------------

Enter any positive integer value: 4
@
4@
56@
789@

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

------------------------- CODE CHANGES FOR 3RD RUN -------------------------
DIAGONAL_CHAR = '='
------------------------------ START OF 3RD RUN ------------------------------

Enter any positive integer value: 1
=

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

------------------------- CODE CHANGES FOR 4TH RUN -------------------------
DIAGONAL_CHAR = '.'
------------------------------ START OF 4TH RUN ------------------------------

Enter any positive integer value: 10
.
0.
12.
345.
6789.
01234.
567890.
1234567.
89012345.
678901234.

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

------------------------- CODE CHANGES FOR 5TH RUN -------------------------
DIAGONAL_CHAR = '+'
```

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

Enter any positive integer value: 6
+
6+
78+
901+
2345+
67890+

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