# Consolidated Assignment 2 Report

This report contains the graded results for the newest of each exercise submitted to the assignment checker prior to 1/26/2022 12:05:59 AM PST.

Student Name: Phillip Ward
Student ID: U09339367
Contact email: phillip.ward@seagate.com
C/C++ Programming I (Section 162461)

Submitted:
- Exercise 0: 1/23/2022 2:07:16 PM PST
- Exercise 1: 1/23/2022 4:22:44 PM PST
- Exercise 2: 1/23/2022 3:36:33 PM PST
- Exercise 3: 1/23/2022 4:07:35 PM PST

Credit to be deducted for uncorrected assignment checker issue(s):
- Exercise 1: 1.2 points (~25%) minimum
- Exercise 2: 0.0 points (0%) minimum
- Exercise 3: 0.0 points (0%) minimum

Score (out of 20 possible): 16

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 avoid credit loss by making corrections and resubmitting. ALL GRADING IS DONE MANUALLY BY THE INSTRUCTOR after the assignment deadline based solely upon the NEWEST submission of each exercise that was submitted BEFORE THE ASSIGNMENT DEADLINE. NO CREDIT will be given for anything submitted after the deadline.

From: Phillip Ward <mailto:phillip.ward@seagate.com>
Subject: C1A2E0_162461_U09339367
Submitted: 1/23/2022 2:07:16 PM PST
Course: C/C++ Programming I (Section 162461)
Student's name: Phillip Ward
Contact email: phillip.ward@seagate.com
Student ID: U09339367
Assignment 2, Exercise 0 (003112338M01005X18112)
Exercise point value: 6
File submitted:
C1A2E0_Quiz.txt

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

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. NO CREDIT will be given for anything submitted after the deadline.

**-3**

Phillip Ward U09339367
Phillip.Ward@seagate.com
C/C++ Programming I
162461 Ray Mitchell
01/23/2022
C1A2E0_Quiz.txt
Quiz Answers

1. A <---E
2. D
3. E <---B
4. B <---D
5. A
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 "Home" page of the course's Canvas website. The assignment checker DOES NOT GRADE your submissions but merely reports on issues so you can avoid credit loss by making corrections and resubmitting. ALL GRADING IS DONE MANUALLY BY THE INSTRUCTOR after the assignment deadline based solely upon the NEWEST submission of each exercise that was submitted BEFORE THE ASSIGNMENT DEADLINE. NO CREDIT will be given for anything submitted after the deadline.

From: Phillip Ward <mailto:phillip.ward@seagate.com>
Subject: C1A2E1_162461_U09339367
Submitted: 1/23/2022 4:22:44 PM PST
Course: C/C++ Programming I (Section 162461)
Student's name: Phillip Ward
Contact email: phillip.ward@seagate.com
Student ID: U09339367
Assignment 2, Exercise 1 (001696742M01005X78696)
Exercise point value: 5
File submitted:
C1A2E1_main.cpp

"Static analysis" results:
1 warning as follows:
**1 compiler warning (Microsoft compiler);**
1 recommendation;
"Runtime" results:
Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);

STANDARD GRADING POLICY:
The MINIMUM deduction is the greater of the following for compile-time issues plus a possible additional deduction for runtime issues, if any:
100% if any "goto" statement is used, else
~45% if any compiler or linker error, else
~25% if any warning, else
~15% if any advisory, else
0% if any recommendation. -1

**C1A2E1: YOUR MINIMUM DEDUCTION: 1.2 points (~25%) To avoid deductions please correct this exercise and resubmit to the assignment checker before the assignment deadline.**

##### The Microsoft compiler found 1 problem. #####
(https://www.visualstudio.com/)

??????????
C1A2E1_main.cpp(27): warning C4244: 'initializing': conversion from 'int' to 'char', possible loss of data
*** EXPLANATION ***
"Loss of data" occurs whenever an expression having an arithmetic type is converted to a different arithmetic type that either cannot fully represent the original range of values or cannot represent them as precisely. As an example of an out of range case, on most implementations the range of type short is smaller than that of type long. While conversions from long to short work fine for values within the range of short, the result of out of range conversions is undefined. As an example of a loss of precision case, on some implementations the number of significant digits that can be represented by type float is less than for type int, even though the value range of type float is much greater. As a result, conversions from int to float work fine if enough of the low-order significant digits are 0s, but precision is lost if they are not. A "loss of data" warning, however, does not mean that data will actually be lost,

because the compiler has no way of knowing the runtime values of variable expressions. Instead, the warning is merely a notification to the programmer that the possibility exists.  It is every programmer's responsibility to know the range and precision of values his/her program is processing and select appropriate data types accordingly.  To suppress this warning in cases where you are certain that loss of data will not actually occur, or that it is not detrimental if it does, cast the expression being converted to the type of the expression it is being converted to.

##### The custom validator found 1 problem. #####
(http://www.MeanOldTeacher.com/AssignmentCheckerKnownIssues.pdf)

??????????
C1A2E1_main.cpp(...) recommendation R151:  "Landscape" page orientation detected
NOTE: There will be no deduction for this but correction is recommended.
*** EXPLANATION ***
Of the 30 total lines in your file, 4 of them (13, 14, 15, and 28) exceed the 80 "portrait" orientation column limit.  As a result "landscape" orientation has been selected instead, which allows 110 columns.  However, portrait is more common and is usually preferred because shorter lines are more readable.  Long lines are often caused by unnecessarily long identifier names, which may or may not be a problem in your code. The recommended 80-column limit can be accidentally exceeded if an editor does not provide a clear indication of which columns characters are in, but when configured properly any good code editor will provide it by either displaying column numbers directly, displaying a vertical column "guideline", highlighting characters that go past a specified column, etc.  Please consult your editor's documentation to see if it can provide a convenient column number indication to help with your code formatting. If it cannot, I recommend using one that can.

1 80 81 <—— Not Recommended ——> 110

```
1  //
2  // Phillip Ward U09339367
3  // Phillip.Ward@seagate.com
4  // C/C++ Programming I
5  // 162461 Ray Mitchell
6  // 01/23/2022
7  // C1A2E1_main.cpp
8  // Win10
9  // g++ 11.2.0
10 //
11 // A C++ program which takes in a char and outputs its lowercase equivalent
12 //
13 // 1. Because ASCII characters are really just numeric values in a lookup table, an input of a non
14 // capital letter will just be the ASCII character that corresponds to its own numeric value minus 32
15 // 2.  It will always print the equivalent as @ because cin.get() only gets the first character and
16 // the lowercase equivalent of a whitspace is the @ sign.
17
18 #include <iostream>
19 using namespace std;
20
21 int main() {
22     int upperToLowerDiff = 'a' - 'A';
23
24     cout << "Enter any character: ";
25     char inputAsChar;
26     cin.get(inputAsChar);
27     char lowercaseEquivalent = inputAsChar + upperToLowerDiff;
28     cout << "The lowercase equivalent of '" << inputAsChar << "' is '" << lowercaseEquivalent << "'\n";
29     return(0);
30 }
```

Unnecessarily long names can result in:
1. autowraps
2. avoidable line splits
3. avoidable landscape page orientations
4. difficult to read code

```
********** 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  **************
************  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 "Home" page of the course's Canvas website. The assignment checker DOES NOT GRADE your submissions but merely reports on issues so you can avoid credit loss by making corrections and resubmitting. ALL GRADING IS DONE MANUALLY BY THE INSTRUCTOR after the assignment deadline based solely upon the NEWEST submission of each exercise that was submitted BEFORE THE ASSIGNMENT DEADLINE. NO CREDIT will be given for anything submitted after the deadline.

From: Phillip Ward <mailto:phillip.ward@seagate.com>
Subject: C1A2E2_162461_U09339367
Submitted: 1/23/2022 3:36:33 PM PST
Course: C/C++ Programming I (Section 162461)
Student's name: Phillip Ward
Contact email: phillip.ward@seagate.com
Student ID: U09339367
Assignment 2, Exercise 2 (002803549M01005X30803)
Exercise point value: 5
File submitted:
C1A2E2_main.c

"Static analysis" results:
1 recommendation;
"Runtime" results:
Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);

STANDARD GRADING POLICY:
The MINIMUM deduction is the greater of the following for compile-time issues plus a possible additional deduction for runtime issues, if any:
100% if any "goto" statement is used, else
~45% if any compiler or linker error, else
~25% if any warning, else
~15% if any advisory, else
0% if any recommendation.

C1A2E2: YOUR MINIMUM DEDUCTION: 0.0 points (0%)

##### The custom validator found 1 problem. #####
(http://www.MeanOldTeacher.com/AssignmentCheckerKnownIssues.pdf)

??????????
C1A2E2_main.c(...) recommendation R151: "Landscape" page orientation detected
NOTE: There will be no deduction for this but correction is recommended.
*** EXPLANATION ***
Of the 32 total lines in your file, 1 of them (24) exceeds the 80 "portrait" orientation column limit. As a result "landscape" orientation has been selected instead, which allows 110 columns. However, portrait is more common and is usually preferred because shorter lines are more readable. Because none of your code exceeds 80 columns, neither should your comments. Long lines are often caused by unnecessarily long identifier names, which may or may not be a problem in your code. The recommended 80-column limit can be accidentally exceeded if an editor does not provide a clear indication of which columns characters are in, but when configured properly any good code editor will provide it by either displaying column numbers directly, displaying a vertical column "guideline", highlighting characters that go past a specified column, etc. Please consult your editor's documentation to see if it can provide a convenient column number indication to help with your code formatting. If it cannot, I recommend using one that can.

1 80 81 <—— Not Recommended ——> 110

```
1  //
2  // Phillip Ward U09339367
3  // Phillip.Ward@seagate.com
4  // C/C++ Programming I
5  // 162461 Ray Mitchell
6  // 01/23/2022
7  // C1A2E2_main.c
8  // Win10
9  // g++ 11.2.0
10 //
11 // A C program which outputs a triangle of characters
12 //
13 #include <stdio.h>
14 #define LEADER_CHAR '^'
15 #define DIAGONAL_CHAR '@'
16
17 int main(void) {
18     int inputInt;
19     printf("input any positive decimal integer:");
20     scanf("%d", &inputInt);
21     //Number of lines will be equal to the input number
22     for (int lineNum = inputInt; lineNum > 0; lineNum--)
23     {
24         //number of leading characters will start at the input number and get succesively smaller
25         for (int spaceNum = 1; spaceNum < lineNum; spaceNum++)
26         {
27             printf("%c", LEADER_CHAR);
28         }
29         printf("%c\n", DIAGONAL_CHAR);
30     }
31     return(0);
32 }
```

If your code doesn't exceed 80 columns, why not also limit comment lengths to 80 columns and avoid landscape mode?

```
********** 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  **************
*************  INSTRUCTOR WILL DO A MORE THOROUGH CHECK DURING  **************
*************                  MANUAL GRADING.                    **************


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

input any positive decimal integer:2
#@
@

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

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

input any positive decimal integer:4
###@
##@
#@
@

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

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

input any positive decimal integer:1
=

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

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

input any positive decimal integer:10
         .
        .
       .
      .
     .
    .
   .
  .
 .
.

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

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

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

input any positive decimal integer:6
++++++
+++++
++++
+++
++
+

------------------------------- 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 "Home" page of the course's Canvas website. The assignment checker DOES NOT GRADE your submissions but merely reports on issues so you can avoid credit loss by making corrections and resubmitting. ALL GRADING IS DONE MANUALLY BY THE INSTRUCTOR after the assignment deadline based solely upon the NEWEST submission of each exercise that was submitted BEFORE THE ASSIGNMENT DEADLINE. NO CREDIT will be given for anything submitted after the deadline.

From: Phillip Ward <mailto:phillip.ward@seagate.com>
Subject: C1A2E3_162461_U09339367
Submitted: 1/23/2022 4:07:35 PM PST
Course: C/C++ Programming I (Section 162461)
Student's name: Phillip Ward
Contact email: phillip.ward@seagate.com
Student ID: U09339367
Assignment 2, Exercise 3 (002519354M01005X83519)
Exercise point value: 4
File submitted:
C1A2E3_main.cpp

"Static analysis" results:
1 recommendation;
"Runtime" results:
Program ran - No errors detected during preliminary testing (SEE ATTACHMENT);

STANDARD GRADING POLICY:
The MINIMUM deduction is the greater of the following for compile-time issues plus a possible additional deduction for runtime issues, if any:
100% if any "goto" statement is used, else
~45% if any compiler or linker error, else
~25% if any warning, else
~15% if any advisory, else
0% if any recommendation.

C1A2E3: YOUR MINIMUM DEDUCTION: 0.0 points (0%)

##### The custom validator found 1 problem. #####
(http://www.MeanOldTeacher.com/AssignmentCheckerKnownIssues.pdf)

??????????
C1A2E3_main.cpp(...) recommendation R151: "Landscape" page orientation detected
NOTE: There will be no deduction for this but correction is recommended.
*** EXPLANATION ***
Of the 34 total lines in your file, 1 of them (26) exceeds the 80 "portrait" orientation column limit. As a result "landscape" orientation has been selected instead, which allows 110 columns. However, portrait is more common and is usually preferred because shorter lines are more readable. Because none of your code exceeds 80 columns, neither should your comments. Long lines are often caused by unnecessarily long identifier names, which may or may not be a problem in your code. The recommended 80-column limit can be accidentally exceeded if an editor does not provide a clear indication of which columns characters are in, but when configured properly any good code editor will provide it by either displaying column numbers directly, displaying a vertical column "guideline", highlighting characters that go past a specified column, etc. Please consult your editor's documentation to see if it can provide a convenient column number indication to help with your code formatting. If it cannot, I recommend using one that can.

1 80 81 <—— Not Recommended ——> 110

```
1  //
2  // Phillip Ward U09339367
3  // Phillip.Ward@seagate.com
4  // C/C++ Programming I
5  // 162461 Ray Mitchell
6  // 01/23/2022
7  // C1A2E3_main.cpp
8  // Win10
9  // g++ 11.2.0
10 //
11 // A simple C++ program that outputs a triangle of characters.
12
13 #include <iostream>
14 using namespace std;
15
16 int main() {
17     const char LEADER_CHAR = '^';
18     const char DIAGONAL_CHAR = '$';
19     cout << "Enter any positive decimal integer: ";
20     int inputAsInt;
21     cin >> inputAsInt;
22
23     //Number of lines will be equal to the input number
24     for (int lineNum = inputAsInt; lineNum > 0; lineNum--)
25     {
26         //number of leading characters will start at the input number and get succesively smaller
27         for (int spaceNum = 1; spaceNum < lineNum; spaceNum++)
28         {
29             cout << LEADER_CHAR;
30         }
31         cout << DIAGONAL_CHAR << "\n";
32     }
33     return(0);
34 }
```

If your code doesn't exceed 80 columns, why not also limit comment lengths to 80 columns and avoid landscape mode?

```
********** 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  **************
************  INSTRUCTOR WILL DO A MORE THOROUGH CHECK DURING  **************
************                   MANUAL GRADING.                    **************


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

Enter any positive decimal integer: 2
#@
@

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

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

Enter any positive decimal integer: 4
###@
##@
#@
@

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

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

Enter any positive decimal integer: 1
=

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

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

Enter any positive decimal integer: 10
         .
        .
       .
      .
     .
    .
   .
  .
 .
.

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

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

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

Enter any positive decimal integer: 6
++++++
+++++
++++
+++
++
+

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