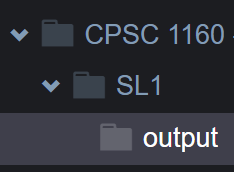
Setting up the programming environment

1. Create a folder, SL1.

|  |  |
| --- | --- |
|  |  |

1. Create a subfolder, output, inside SL1



1. Save all your .cpp files inside SL1 folder
2. To compile, make sure you are inside SL1 folder



1. To run the program

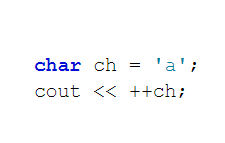


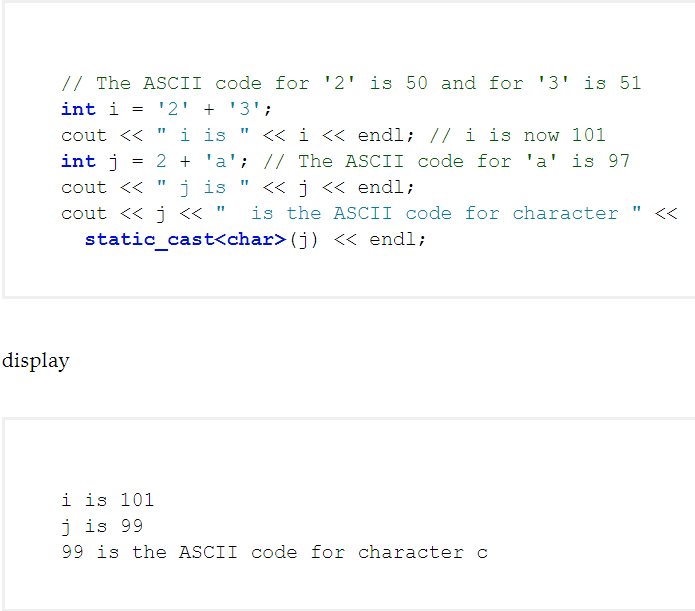
1. What to submit

Put all .cpp files in 1 zip file then upload to D2L

Q1. [2 marks]

|  |  |
| --- | --- |
|  |  |





Will display b

Task: Complete the code

Filename: q1\_yourstudentid.cpp

// your name

#include <iostream>

using namespace std;

int main()

{

char lowercaseLetter;

cout << "Enter a lowercase letter: ";

cin >> lowercaseLetter;

char uppercaseLetter =

static\_cast<char>(**FILL\_CODE here**);

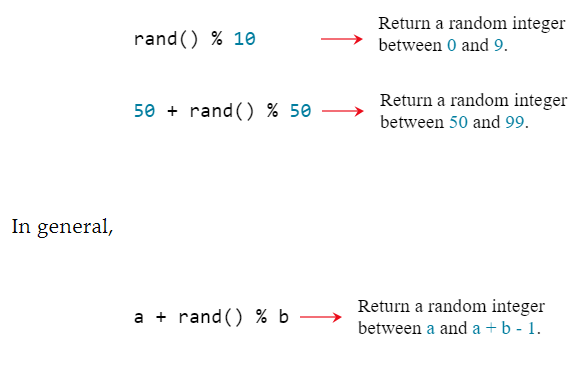
cout << "The corresponding uppercase letter is "

<< uppercaseLetter << endl;

return 0;

}

Q2. [2 marks]



Task: Ask the user to enter a starting and ending character then generate a random letter between the two characters.

Filename: q2\_yourstudentid.cpp

Example:

Enter a starting character: a

Enter an ending character: f

The random character between a and f is e

// Your name

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main()

{

cout << "Enter a starting character: ";

char startChar;

cin >> startChar;

cout << "Enter an ending character: ";

char endChar;

cin >> endChar;

// Get a random character

srand(time(0));

char randomChar = **FILL\_CODE here**

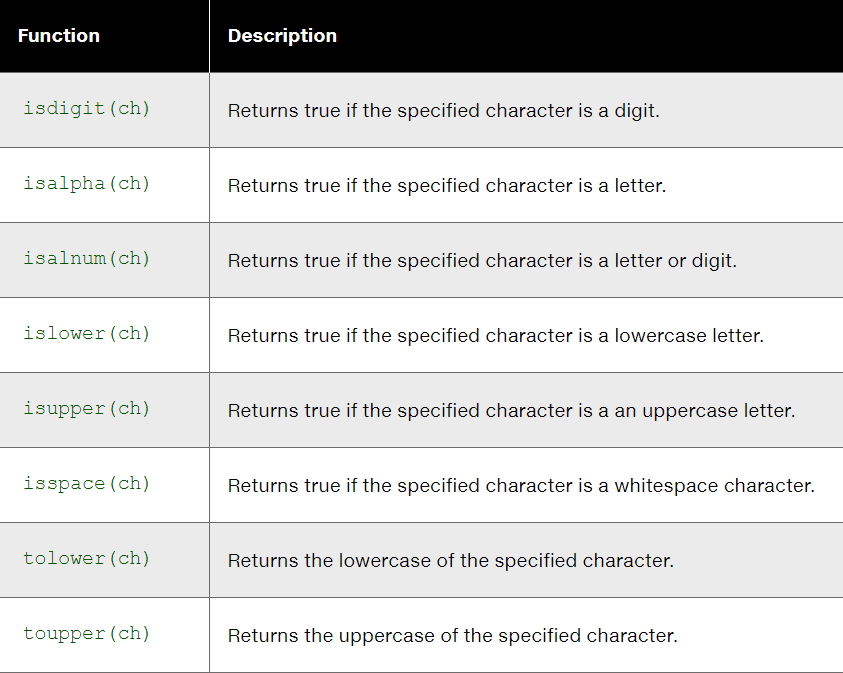
cout << "The random character between " << startChar << " and "

<< endChar << " is " << randomChar << endl;

return 0;

}

Q3. [2 marks]



Task: Complete the code below using the Character functions

Filename: q3\_yourStudentid.cpp

// your name

#include <iostream>

#include <cctype>

using namespace std;

int main()

{

cout << "Enter a character: ";

char ch;

cin >> ch;

cout << "You entered " << ch << endl;

if (islower(ch))

{

cout << "It is a lowercase letter " << endl;

cout << "Its equivalent uppercase letter is " << **FILL\_CODE here** << endl;

}

else if (isupper(ch))

{

cout << "It is an uppercase letter " << endl;

cout << "Its equivalent lowercase letter is " << **FILL\_CODE here** << endl;

}

else if **FILL\_CODE here**

{

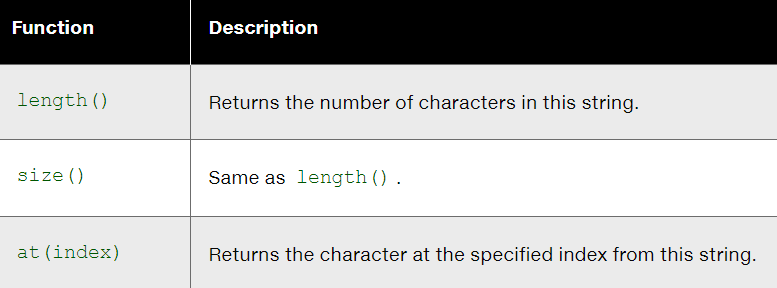
cout << "It is a digit character " << endl;

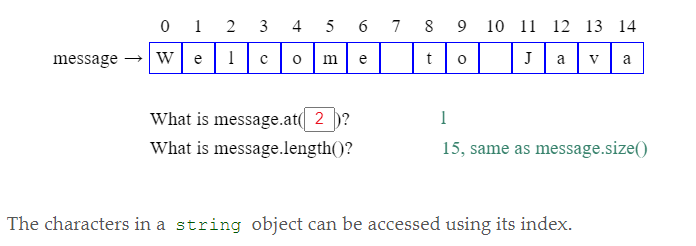
}

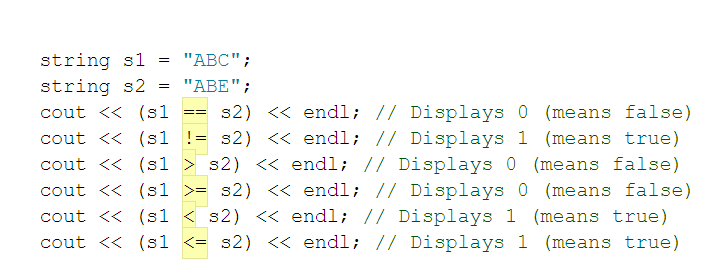
return 0;

}

Q4. [2 marks]







|  |  |
| --- | --- |
| Note: input ends with a whitespace character | Note: default 3rd argument is \n |

Task: Prompt the user to enter two cities and display them alphabetically

Filename: q4\_yourStudentid.cpp

// your name

#include <iostream>

#include <string>

using namespace std;

int main()

{

string city1, city2;

cout << "Enter the first city: ";

**FILL\_CODE here**

cout << "Enter the second city: ";

**FILL\_CODE here**

cout << "The cities in alphabetical order are ";

if (**FILL\_CODE here**)

cout << city1 << " " << city2 << endl;

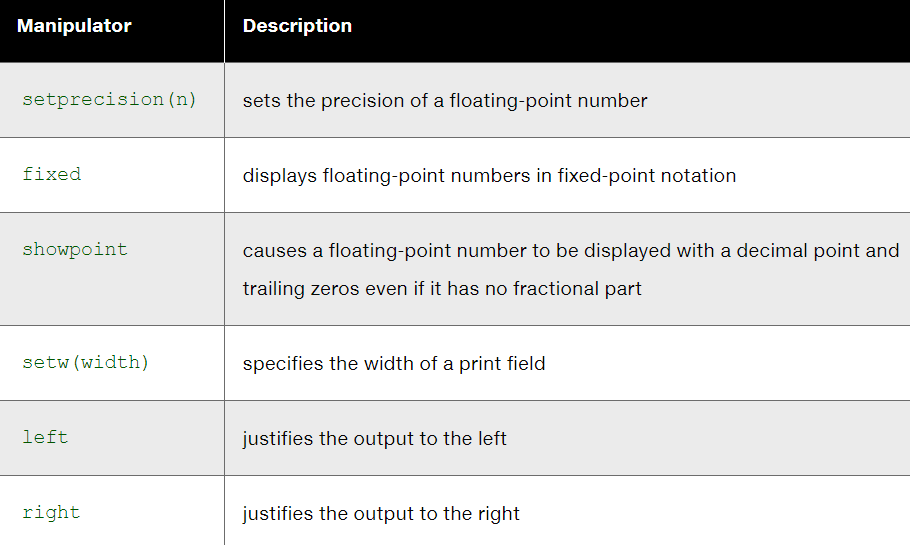
else

cout << city2 << " " << city1 << endl;

return 0;

}

Q5. [2 marks]

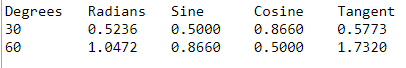


|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

Task: Complete the code and the display the data in each formatted output

Filename: q5\_yourstudentid.cpp

Sample output



// your name

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

int main()

{

// Display the header of the table

cout << left << setw(10) << "Degrees" << setw(10) << "Radians"

<< setw(10) << "Sine" << setw(10) << "Cosine" << setw(10)

<< "Tangent" << endl;

Formula:

* sin(radians)
* cos(radians)
* tan(radians)

// Display values for 30 degrees

const double PI = 3.14159;

double degrees = 30;

double radians = degrees \* (PI / 180);

cout << setw(10) << degrees << **Your\_CODE\_HERE** << endl;

// Display values for 60 degrees

degrees = 60;

double radians = degrees \* (PI / 180);

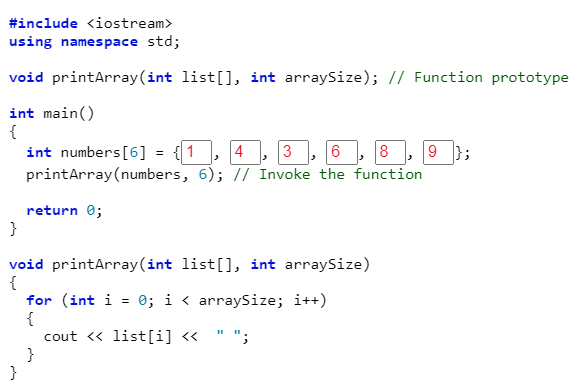
cout << setw(10) **Your\_CODE\_HERE** <<endl;

return 0;

}

Q6. [3 marks]

|  |  |  |
| --- | --- | --- |
|  |  |  |



Passing an array to a function

Task:

Filename: q6\_yourstudentid.cpp

// your name

#include <iostream>

using namespace std;

const int TOTALGRADES = 10; // TOTALGRADES is the maximum size of the array

// function prototypes

void getData(int array[], int& sizeOfArray); // will read values into the array

float findAverage(const int array[], int sizeOfArray); // compute avg

int main(){

int grades[TOTALGRADES]; // defines an array that holds up to 10 integers

int numberOfGrades = 0; // the number of grades read

float average; // the average of all grades read

getData(grades, numberOfGrades); // getData is called to read the grades into

// the array and store how many grades there

// are in numberOfGrades

average = findAverage(grades, numberOfGrades);

cout << endl << "The average of the " << numberOfGrades

<< " grades read in is " << average << "." << endl << endl;

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// getData

//

// task: This function inputs and stores data in the grades array.

// data in: none (the parameters contain no information needed by the getData function)

// data out: an array containing grades and the number of grades

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void getData(int array[], int& sizeOfArray)

{

int pos = 0; // array index which starts at 0

int grade; // holds each individual grade read in

cout << "Please input a grade or type -99 to stop: " << endl;

**Your\_CODE\_HERE**

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// findAverage

//

// task: This function finds and returns the average of the values

// data in: the array containing grades and the array size

// data returned: the average of the grades contained in that array

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

float findAverage (const int array[], int sizeOfArray)

{

int sum = 0; // holds the sum of all grades in the array

**Your\_CODE\_HERE**

}

Q7. [3 marks]

|  |  |  |
| --- | --- | --- |
|  |  |  |
| How to initialize 2D with random numbers | | |
| How to initialize 2D with user inputs | | |
| How to create a function that accepts 2D array  const is optional (if you do not want the array to be modified)  void displayPage(const char p[][100], int sizeDimension1) {  for (int index1 = 0; index1 < sizeDimension1; index1++) {  //Printing one line:  for (int index2 = 0; index2 < 100; index2++)  cout << p[index1][index2]; cout << endl;  }  } | | |

Task: Write a program that will allow two users to play tic-tac-toe. The program should ask for moves alternately from player X and player O.

Filename: q7\_studentid.cpp

Must use 2D array to represent the tic-tac-toe board

|  |  |
| --- | --- |
| Sample display | The players enter their moves by entering the position number they wish to mark. After each move, the program displays the changed board.  You can use cin to get user input.  The game terminates either a winner has been declared or a tie |

Q8. [3 marks]

|  |  |
| --- | --- |
|  |  |

<https://www.cplusplus.com/reference/string/string/c_str/>

<http://www.cplusplus.com/doc/tutorial/structures/>

|  |  |
| --- | --- |
|  |  |

Task: Write an application that would prompt the user to enter 5 employees.

Other requirements:

1. Create an array out of struct Employee (see the screenshot above)
2. Create a function getData. This function gets the user input and store the data into the array.
   * At the very least, the function should have the struct employee array and array size as parameters
3. Create a function displayData. This function displays the contents of the array
   * Parameters must include struct employee array and array size
   * Formatted output (just a sample below with 3 employees)

