UWE
Bristol

The
British
College

WORKSHEET 1

24030183

SUBMITTED BY: Sujal Adhikari
STUDENT ID: 24030183

INTRODUCTION

This documentation was written as part of my C++ assessment. This project consisted of some exercises that I did to learn the fundamentals of C++ programming, like manipulating data types, using conditionals, loops, and arrays. Some of these programs included a temperature conversion calculator, a random number generator for guessing games with 10 different difficulty levels, an Array Sorter, a cinema ticket booking system, etc. Note: This task helped me with error handling and validation of user input, one of the utmost important programming practice.

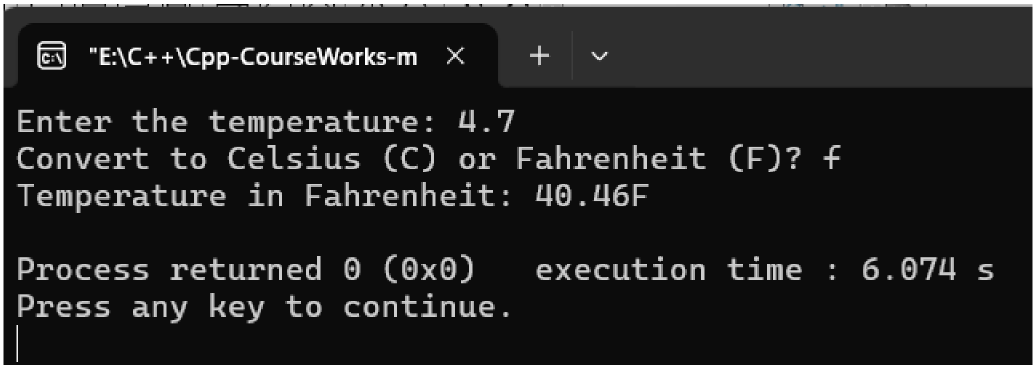
QUESTION 1.1

1. Write a program that takes a temperature value from the user. It should then allow the user to choose between Celsius (C) and Fahrenheit (F) for conversion. After the user selection, it should then convert the entered temperature to the chosen scale and display the result.
   Use appropriate data types for temperature and handle error like non-numeric input.
   Use the following formula for conversion:
   F = (C x 9/5) + 32
   C = (F - 32) x 5/9

Code Implementation

```
//1.1-Write a program that takes a temperature value from the user.

#include <iostream>

using namespace std;

int main() {

  double temp;

  char choice;

  cout << "Enter the temperature: ";

  cin >> temp;

  cout << "Convert to Celsius (C) or Fahrenheit (F)? ";

  cin >> choice;

  if (choice == 'C' || choice == 'c') {

    double celsius = (temp - 32) * 5 / 9;

    cout << "Temperature in Celsius: " << celsius << "C" << endl;

  }

  else if (choice == 'F' || choice == 'f') {

    double fahrenheit = (temp * 9 / 5) + 32;

    cout << "Temperature in Fahrenheit: " << fahrenheit << "F" << endl;
}
  else {
    cout << "Invalid choice! Please enter 'C' or 'F'." << endl;
  }
  return 0;
}
```

Output



```
Enter the temperature: 4.7
Convert to Celsius (C) or Fahrenheit (F)? f
Temperature in Fahrenheit: 40.46F

Process returned 0 (0x0)   execution time : 6.074 s
Press any key to continue.
```

QUESTION 1.2

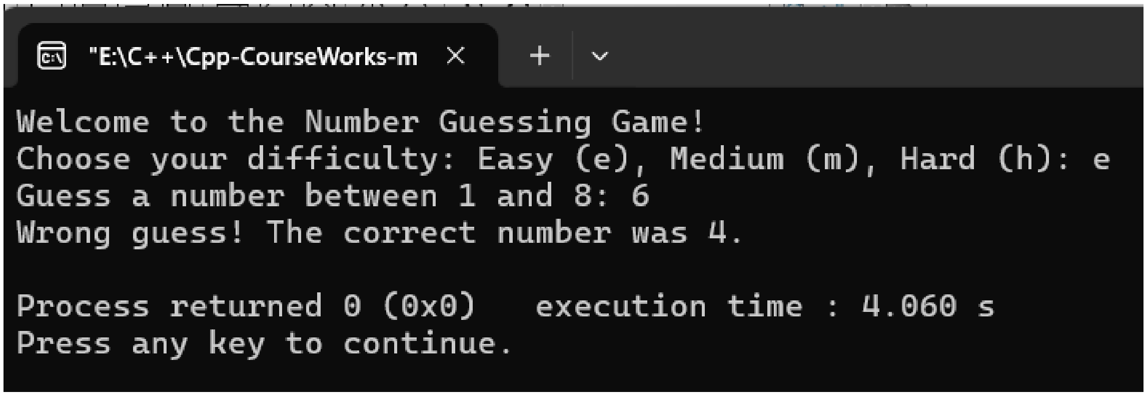
1. Write a C++ program to implement a number guessing game with different difficulty levels. Easy difficulty ranges from 1-8, medium from 1-30, hard from 1-50.Then,generate a random number to check if the guess is correct based on the user's selection.

Code Implementation

```
#include <iostream>
#include <ctime>  //generate random numbers
using namespace std;
int main() {
srand(time(0));  //random number generator
int easy = rand() % 8 + 1;    //Random number between 1 and 8
int medium = rand() % 30 + 1; //Random number between 1 and 30
int hard = rand() % 50 + 1;   //Random number between 1 and 50
int guess;
char difficulty;
cout << "Welcome to the Number Guessing Game!" << endl;
cout << "Choose your difficulty: Easy (e), Medium (m), Hard (h): ";
cin >> difficulty;
switch (difficulty) {
case 'e': case 'E':
cout << "Guess a number between 1 and 8: ";
cin >> guess;
if (guess == easy) {
cout << "Congratulations! You guessed the correct number." << endl;
} else {
cout << "Wrong guess! The correct number was " << easy << "." << endl;
}
break;
case 'm': case 'M':
```

```
cout << "Guess a number between 1 and 30: ";
      cin >> guess;
      if (guess == medium) {
        cout << "Congratulations! You guessed the correct number." << endl;
      } else {
        cout << "Wrong guess! The correct number was " << medium << "." << endl;
      }
      break;
    case 'h': case 'H':
      cout << "Guess a number between 1 and 50: ";
      cin >> guess;
      if (guess == hard) {
        cout << "Congratulations! You guessed the correct number." << endl;
      } else {
        cout << "Wrong guess! The correct number was " << hard << "." << endl;
      }
      break;
    default:
      cout << "Invalid input! Please restart the game and choose a valid difficulty." << endl;
  }
  return 0;
}
```

Output



```
Welcome to the Number Guessing Game!
Choose your difficulty: Easy (e), Medium (m), Hard (h): e
Guess a number between 1 and 8: 6
Wrong guess! The correct number was 4.

Process returned 0 (0x0)   execution time : 4.060 s
Press any key to continue.
```

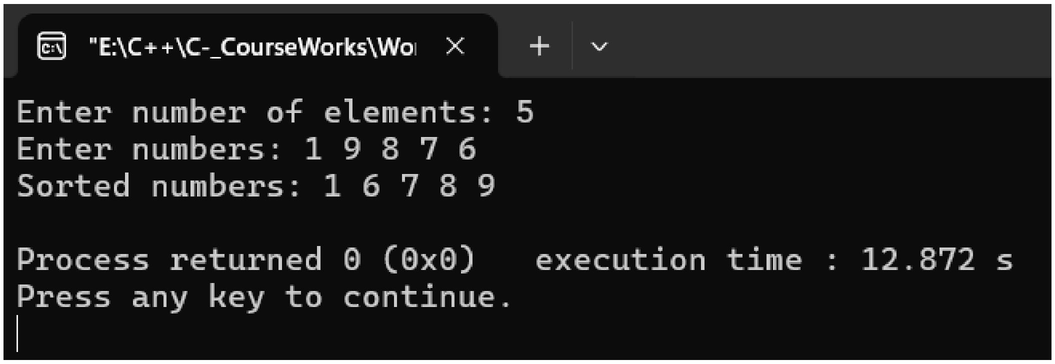
QUESTION 1.3

1. Write a program that reads an array of integer numbers from the user and sorts the numbers in the ascending order.

Code Implementation

```
#include <iostream>

#include <algorithm>  //built-in sort() function

using namespace std;

int main() {

  int n, arr[100];  //Declare an array to store numbers max size 100

  cout << "Enter number of elements: ";

  cin >> n;

  if (n > 100 || n <= 0) {

    cout << "Invalid input! Please enter a number between 1 and 100." << endl;

    return 1;  //Exit the program with an error code

  }

  cout << "Enter numbers: ";
for (int i = 0; i < n; i++) {

cin >> arr[i];

}

sort(arr, arr + n);  //Sort the array in ascending order using the built-in sort() function

cout << "Sorted numbers: ";

for (int i = 0; i < n; i++) {

cout << arr[i] << " ";

}

cout << endl;

return 0;

}
```

Output



```
Enter number of elements: 5
Enter numbers: 1 9 8 7 6
Sorted numbers: 1 6 7 8 9

Process returned 0 (0x0)   execution time : 12.872 s
Press any key to continue.
```

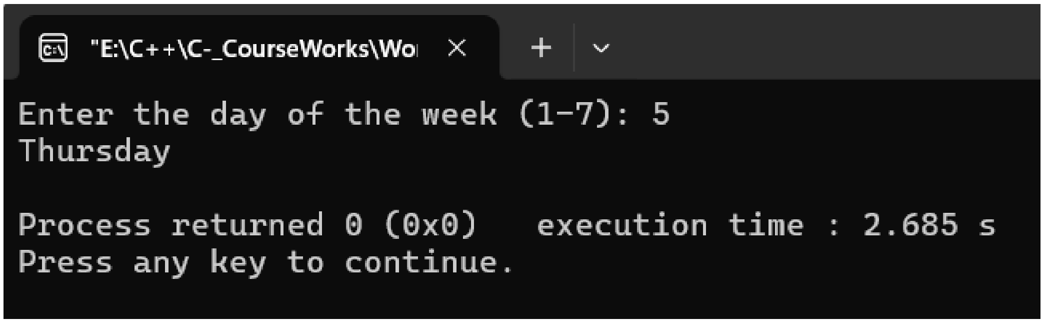
QUESTION 1.4

1. Write a program that reads a number from the user and based on the user input, it says what day of the week it is, Sundays being 1 and Saturdays being 7. You system should give appropriate response for invalid input entries.

Code Implementation

```
#include <iostream>

using namespace std;

int main() {

  int day;

  cout << "Enter the day of the week (1-7): ";

  cin >> day;

  switch (day) {

    case 1:

      cout << "Sunday" << endl;

      break;

    case 2:

      cout << "Monday" << endl;

      break;

    case 3:

      cout << "Tuesday" << endl;

      break;

    case 4:

      cout << "Wednesday" << endl;

      break;

    case 5:

      cout << "Thursday" << endl;

      break;

    case 6:

      cout << "Friday" << endl;

      break;
case 7:
      cout << "Saturday" << endl;
      break;
    default:
      cout << "Invalid day of the week! Please enter a number between 1 and 7." << endl;
  }
  return 0;
}
```

Output



```
Enter the day of the week (1-7): 5
Thursday

Process returned 0 (0x0)   execution time : 2.685 s
Press any key to continue.
```

QUESTION 2.1

1. Create a program that takes a positive integer as input and determines whether it's a "bouncy number". A bouncy number is one where the digits neither consistently increase nor consistently decrease when read from left to right. For example:

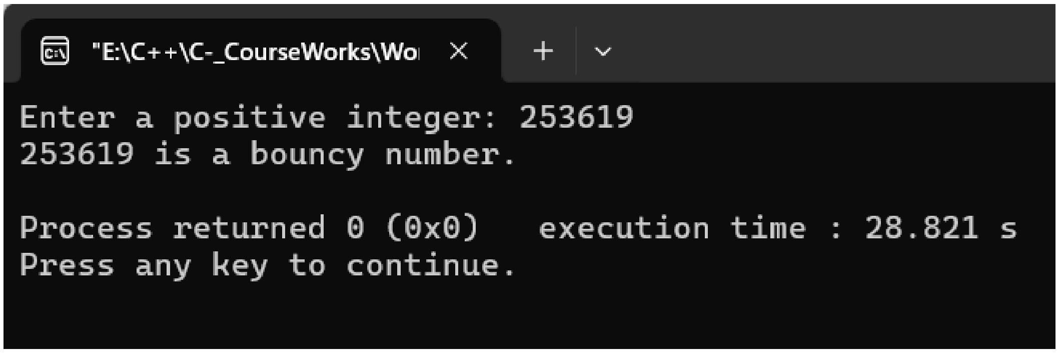
- 123 is NOT bouncy (digits consistently increase)
- 321 is NOT bouncy (digits consistently decrease)
- 120 is bouncy (neither consistently increasing nor decreasing)

Code Implementation

```
#include <iostream>
using namespace std;
bool isBouncy(int num) {
    if (num < 100) return false;  //Numbers below 100 are NOT bouncy
    bool increasing = false, decreasing = false;
    int lastDigit = num % 10;  //Extract last digit
    num /= 10;  //Remove last digit
    while (num > 0) {
        int currentDigit = num % 10;  //Extract next digit
        if (currentDigit < lastDigit) increasing = true;   //Checking increasing trend
        if (currentDigit > lastDigit) decreasing = true;   //Checking decreasing trend
        if (increasing && decreasing) return true;  //If both are true, it's bouncy
        lastDigit = currentDigit;  //Move to the next digit
        num /= 10;  //Remove last digit
    }
    return false;  //Not a bouncy number
}
int main() {
    int num;
    cout << "Enter a positive integer: ";
    cin >> num;
    if (isBouncy(num)) {
        cout << num << " is a bouncy number." << endl;
```

```
  } else {
     cout << num << " is NOT a bouncy number." << endl;
  }
  return 0;
}
```

Output



```
Enter a positive integer: 253619
253619 is a bouncy number.

Process returned 0 (0x0)   execution time : 28.821 s
Press any key to continue.
```

QUESTION 3.1

1. Write a program that manages a cinema ticket booking system. The program should display a 5x5 seating arrangement where:

   1. Available seats are marked with 'O'
   2. Booked seats are marked with 'X'

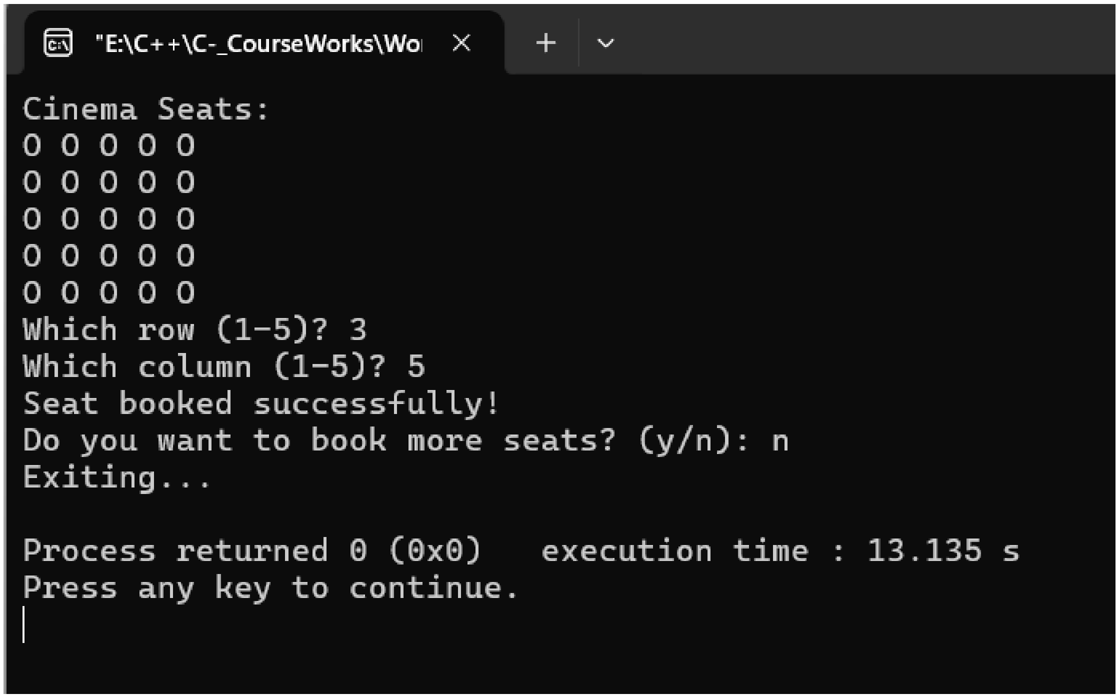
Program should:

1. Display the current seating arrangement
2. Ask user for row and column number (1-5) for booking
3. Mark that seat as booked ('X')
4. Show updated seating after each booking
5. Display error if user selects already booked seat
6. Display error if user enters invalid row/column numbers

Code Implementation

```
#include <iostream>
using namespace std;
int main() {
    char seats[5][5] = {  // Initialize a 5x5 seating arrangement
        {'O', 'O', 'O', 'O', 'O'},
        {'O', 'O', 'O', 'O', 'O'},
        {'O', 'O', 'O', 'O', 'O'},
        {'O', 'O', 'O', 'O', 'O'},
        {'O', 'O', 'O', 'O', 'O'}
    };
    while (true) { //loop for seat booking
        cout << "Cinema Seats:\n";
        for (int row = 0; row < 5; row++) {  //Display the current seat
            for (int col = 0; col < 5; col++) {
                cout << seats[row][col] << " ";  //Print seat status
            }
            cout << endl;
        }
        cout << "Which row (1-5)? ";
        int row;
        cin >> row;
        cout << "Which column (1-5)? ";
        int col;
        cin >> col;
if (row == 0 && col == 0) {
      cout << "Exiting...\n";
      break;
    }
    if (row < 1 || row > 5 || col < 1 || col > 5) {  //Ensure row and column are within valid range
      cout << "Invalid seat selection! Please enter values between 1 and 5.\n";
      continue;
    }
    // Convert to zero-based indexing
    row--;
    col--;
    if (seats[row][col] == 'X') { //Check if the seat is already booked
      cout << "Sorry, that seat is already taken. Try another.\n";
      continue;
    }
    seats[row][col] = 'X'; //Mark the seat as booked
    cout << "Seat booked successfully!\n";
    char choice;
    cout << "Do you want to book more seats? (y/n): ";
    cin >> choice;
    if (choice == 'n' || choice == 'N') {
      cout << "Exiting...\n";
      break;
    }
  }
  return 0;
}
```

Output



```
Cinema Seats:
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Which row (1-5)? 3
Which column (1-5)? 5
Seat booked successfully!
Do you want to book more seats? (y/n): n
Exiting...

Process returned 0 (0x0)   execution time : 13.135 s
Press any key to continue.
```

CONCLUSION

Overall, this C++ assessment has provided an excellent opportunity for me to learn. This taught me how to use basic programming features like control structures and arrays in real-world problem-solving. Different tasks have honed my skills in writing clean and efficient code handling errors and debugging respectively. Overall, this project helped reinforce my understanding of C++ fundamentals and gave me greater confidence to undertake more advanced programming projects in the future.