

Worksheet 1

-Utsab Poudel

Task 1 : Programming Exercises:[Data types and Conditional Statements]

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:

$$F = (C \times 9/5) + 32$$

$$C = (F - 32) \times 5/9$$
 [10 marks]

```
#include <iostream>
using namespace std;

class Temperature
{
private:
    float temp;
    char scale;

public:
    //function to get temperature from user
    void gettemp()
    {
        cout << "Enter the temperature: ";
        cin >> temp;
    }
    //function to get scale from user
    void getscale()
    {
        cout << "Pressing C will convert celsius into fahrenheit " << endl <<
        "Pressing F will covert fahrenheit into celsius" << endl;
        cin >> scale;
        if (scale != 'C' && scale != 'F' && scale != 'c' && scale != 'f')
        {
            cout << "Invalid input" << endl;
            getscale();
        }
    }
    //function to convert temperature
    void conversion()
```

```

    {
        if (scale == 'C' || scale == 'c')
        {
            cout << "Temperature in Fahrenheit: " << (temp * 9 / 5) + 32 << " F"
<< endl;
        }
        else
        {
            cout << "Temperature in Celsius: " << (temp - 32) * 5 / 9 << " C" <<
endl;
        }
    }
};

int main()
{
    Temperature t1;
    t1.gettemp();
    t1.getscale();
    t1.conversion();
    return 0;
}

```

Output:

```

Microsoft Visual Studio Debug Console
Enter the temperature: 12
Pressing C will convert celsius into fahrenheit
Pressing F will covert fahrenheit into celsius
c
Temperature in Fahrenheit: 53.6 F

Enter the temperature: 200
Pressing C will convert celsius into fahrenheit
Pressing F will covert fahrenheit into celsius
f
Temperature in Celsius: 93.3333 C

```

- 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.

[10 marks]

```

#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

class Game {
    int maxRange, actualno, guessno;
public:
    void setdifficulty() {
        int choice;
        cout << "Select Difficulty Level:"<<endl;
        cout << "1. Easy (1-8)"<<endl;
        cout << "2. Medium (1-30)"<<endl;
        cout << "3. Hard (1-50)"<<endl;
        cout << "Enter choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                maxRange = 8;
                break;
            case 2:
                maxRange = 30;
                break;
            case 3:
                maxRange = 50;
                break;
            default:
                cout << "Invalid choice! Please choose again\n";
                setdifficulty();
        }
    }

    void randomno() {
        //this is to give random no
        srand(time(0));

        actualno = rand() % maxRange + 1; // Generates number in range 1 to
maxRange
    }

    void playGame() {
        cout << "Guess a number between 1 and " << maxRange << ":\n";

        do {
            cout << "Enter your guess: ";
            cin >> guessno;

            if (guessno < actualno) {
                cout << "Too low! Try again.\n";
            }
            else if (guessno > actualno) {
                cout << "Too high! Try again.\n";
            }
            else {

```

```
        cout << "Congratulations! You guessed the correct number: " <<
actualno << endl;
    }
    } while (guessno != actualno);
};

int main() {
    Game g1;
    g1.setdifficulty();
    g1.randomno();
    g1.playGame();
    return 0;
}
```

Output:

```
Select Difficulty Level:
1. Easy (1-8)
2. Medium (1-30)
3. Hard (1-50)
Enter choice: 1
Guess a number between 1 and 8:
Enter your guess: 5
Congratulations! You guessed the correct number: 5
```

```

rGu Select Difficulty Level:
Ran 1. Easy (1-8)
    2. Medium (1-30)
    3. Hard (1-50)
lec Enter choice: 3
ch  Guess a number between 1 and 50:
t < Enter your guess: 43
t < Too high! Try again.
t < Enter your guess: 23
t < Too low! Try again.
t < Enter your guess: 34
>> Too high! Try again.
tch Enter your guess: 20
e 1 Too low! Try again.
e 2 Enter your guess: 30
e 3 Too low! Try again.
aul Enter your guess: 34
co Too high! Try again.
ma Enter your guess: 33
    Too high! Try again.
    Enter your guess: 32
    Too high! Try again.
ner Enter your guess: 31
    Congratulations! You guessed the correct number: 31

```

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

[10 marks]

```

#include <iostream>
using namespace std;

class SortArray {
private:
    int size;
    int arr[100];

public:
    void inputArr() {
        //size can be increased as per the requirement for now 100 is used
        cout << "Enter the size of the array (max 100): ";
        cin >> size;
    }
};

```

```

        if (size > 100 || size <= 0) {
            int a;
            throw(a = 1);
        }

        cout << "Enter the elements of the array: ";
        for (int i = 0; i < size; i++) {
            cin >> arr[i];
        }
    }

    void sortArr() {

        for (int i = 0; i < size - 1; i++) {
            for (int j = i + 1; j < size; j++) {
                if (arr[i] > arr[j]) {

                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;

                }
            }
        }
    }

    void displayArr() {
        cout << "The sorted array is: ";
        for (int i = 0; i < size; i++) {
            cout << arr[i] << " ";
        }
        cout << endl;
    }
};

int main() {
    SortArray s1;
    try {
        s1.inputArr();
    }
    catch (int a) {
        cout << "Invalid size of the array" << endl;
        return 0;
    }

    s1.sortArr();
    s1.displayArr();
    return 0;
}

```

Output:

```
Enter the size of the array (max 100): 4
Enter the elements of the array: 1
2
77
42
The sorted array is: 1 2 42 77
```

```
Enter the size of the array (max 100): 6
Enter the elements of the array: 50
23
1
99
20
44
The sorted array is: 1 20 23 44 50 99
```

```
Enter the size of the array (max 100): 500
Invalid size of the array
```

4. 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. Your system should give appropriate response for invalid input entries. **[20 marks]**

```
#include <iostream>
using namespace std;

class dayOfWeek {
private:
    int day;
public:
    void setday() {
        cout << "Enter a number between 1 and 7: ";
        cin >> day;
    }

    void weekday() {
        if (day == 1) {
            cout << "Sunday" << endl;
        }
    }
}
```

```

    }
    else if (day == 2) {
        cout << "Monday" << endl;
    }
    else if (day == 3) {
        cout << "Tuesday" << endl;
    }
    else if (day == 4) {
        cout << "Wednesday" << endl;
    }
    else if (day == 5) {
        cout << "Thursday" << endl;
    }
    else if (day == 6) {
        cout << "Friday" << endl;
    }
    else if (day == 7) {
        cout << "Saturday" << endl;
    }
    else {
        cout << "Invalid input" << endl;
    }
}

void menu() {
    setday();
    weekday();
}

};

int main() {
    dayOfWeek d1;
    d1.menu();
    return 0;
}

```

Output:

```

Enter a number between 1 and 7: 5
Thursday

```

```

Enter a number between 1 and 7: c
Invalid input

```

Task 2: Programming Exercises:[Control Statements]

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)

[25 marks]

```
#include <iostream>
using namespace std;

class BouncyNumber {
    int number;
    //this bool var checks whether its bouncy or not
    bool isBouncy;

public:

    //function to enter number
    void enterno() {
        cout << "Enter a number: ";
        cin >> number;
    }

    //function to check bounce no or not
    void determineBouncyNumber() {
        int temp = number;
        bool increasing = false, decreasing = false;

        int lastno = temp % 10;
        temp /= 10;

        while (temp > 0) {
            int currentno = temp % 10;

            if (currentno < lastno) {
                increasing = true;
            }
            else if (currentno > lastno) {
                decreasing = true;
            }

            lastno = currentno;
            temp /= 10;

            // If both conditions are met, it's bouncy
            if (increasing && decreasing) {
                isBouncy = true;
                break; // breaks loop
            }
        }
    }
};
```

```

    }
}

// If we exit the loop without breaking, it's not bouncy
if (!(increasing && decreasing)) {
    isBouncy = false;
}
}

void checkAndDisplay() {
    enterNo();
    determineBouncyNumber();

    if (isBouncy) {
        cout << "The number is a bouncy number." << endl;
    }
    else {
        cout << "The number is not a bouncy number." << endl;
    }
}
};

int main() {
    BouncyNumber b1;
    b1.checkAndDisplay();
    return 0;
}

```

Output:

The image shows two screenshots of the Microsoft Visual Studio Debug Console. The first screenshot displays the input 'Enter a number: 1209' and the output 'The number is a bouncy number.' The second screenshot displays the input 'Enter a number: 123' and the output 'The number is not a bouncy number.'

Task 3: Programming Exercises on Arrays

1. Write a program that manages a cinema ticket booking system. The program should display a 5x5 seating arrangement where: **[25 marks]**

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

```
#include <iostream>
using namespace std;

class Cinema {
private:
    char seats[5][5];
    bool isValid;
    bool isAvailable;
    bool keepBooking;

    // Show the seating arrangement
    void showSeats() {
        cout << "\nCurrent Seating Arrangement:\n";
        cout << "  1 2 3 4 5\n";
        for (int i = 0; i < 5; ++i) {
            cout << i + 1 << " ";
            for (int j = 0; j < 5; ++j) {
                cout << seats[i][j] << " ";
            }
            cout << endl;
        }
    }

    // Check if input is valid or not
    void checkInput(int row, int col) {
        isValid = (row >= 1 && row <= 5 && col >= 1 && col <= 5);
    }

    // Check if the seat is not already booked
    void checkSeat(int row, int col) {
        isAvailable = (seats[row - 1][col - 1] == '0');
    }

    // Mark seat as booked
    void book(int row, int col) {
        seats[row - 1][col - 1] = 'X';
    }

    //asks user if user wants to book another seat
```

```

void askAgain() {
    char choice;
    cout << "Do you want to book another seat? (y/n): ";
    cin >> choice;
    keepBooking = (choice == 'y' || choice == 'Y');
}

public:
    // Constructor to initialize all seats as available
    Cinema() {
        for (int i = 0; i < 5; ++i)
            for (int j = 0; j < 5; ++j)
                seats[i][j] = '0';
    }

    // Booking menu
    void start() {
        cout << "Welcome to the Cinema Ticket Booking System \n";

        while (true) {
            showSeats();

            int row, col;
            cout << "Enter row number (1-5): ";
            cin >> row;
            cout << "Enter column number (1-5): ";
            cin >> col;

            checkInput(row, col);
            if (!isValid) {
                cout << "Invalid input\n";
                continue;
            }

            checkSeat(row, col);
            if (!isAvailable) {
                cout << "Seat already booked\n";
                continue;
            }

            book(row, col);
            cout << "Seat booked successfully!\n";

            askAgain();
            if (!keepBooking) {
                cout << "Thank you for using the Cinema Ticket Booking System.\n";
                break;
            }
        }
    }
};

int main() {
    Cinema c;
    c.start();
    return 0;
}

```

Output:

```
Welcome to the Cinema Ticket Booking System

Current Seating Arrangement:
  1 2 3 4 5
1 0 0 0 0 0
2 0 0 0 0 0
3 0 0 0 0 0
4 0 0 0 0 0
5 0 0 0 0 0
Enter row number (1-5): 0
Enter column number (1-5): 0
Invalid input

Current Seating Arrangement:
  1 2 3 4 5
1 0 0 0 0 0
2 0 0 0 0 0
3 0 0 0 0 0
4 0 0 0 0 0
5 0 0 0 0 0
Enter row number (1-5): 4
Enter column number (1-5): 5
Seat booked successfully!
Do you want to book another seat? (y/n): y

Current Seating Arrangement:
  1 2 3 4 5
1 0 0 0 0 0
2 0 0 0 0 0
3 0 0 0 0 0
4 0 0 0 0 X
5 0 0 0 0 0
Enter row number (1-5): 3
Enter column number (1-5): 3
Seat booked successfully!
Do you want to book another seat? (y/n): n
Thank you for using the Cinema Ticket Booking System.
```

```
Current Seating Arrangement:
```

```
1 2 3 4 5  
1 0 0 0 X 0  
2 0 0 0 0 0  
3 0 0 0 0 0  
4 0 0 0 0 0  
5 0 0 0 0 0
```

```
Enter row number (1-5): 1
```

```
Enter column number (1-5): 1
```

```
Seat booked successfully!
```

```
Do you want to book another seat? (y/n): n
```

```
Thank you for using the Cinema Ticket Booking System.
```

Task 4

- Check and commit all your solutions in your github repo.
- This task carries no marks but it is mandatory. Ensure that your solution is visible to us.