 Computer Science and Creative Technologies

**Coursework or Assessment Specification**

## Module Details

|  |  |
| --- | --- |
| **Module Code** | UFCFGL-30-1 |
| **Module Title** | Programming in C++ |
| **Module Leader** | Benedict Gaster |
| **Module Tutors** | Aashish Acharya |
| **Year** | 2024-2025 |
| **Component/Element Number** | Coursework |
| **Weighting** | 10% |
| **Element Description** |  |

## Dates

|  |  |
| --- | --- |
| **Date issued to students** | 10/02/2025 |
| **Submission Date** | 10/04/2025 |
| **Submission Place** | Backboard |
| **Submission Time** | 00:00 |
| **Submission Notes** | Submit Gitlab URL |

## Feedback

|  |  |
| --- | --- |
| **Feedback provision will be** | Via Blackboard |

**Uwe id = 23045166**

**Worksheet-1: C++ Basics**

The coursework-1 is based on basics of C++ like data types, control structures and arrays. You are required to go through the lectures and tutorials of the week1, week2, week3 and week4 learning materials as well as reference URLs mentioned in the handout notes.

The marking scheme for this session is as follows:

* Task 1: 50 marks
* Task 2: 25 marks
* Task 3: 25 marks
* Task 4: 0 marks

**Submission Details in the VLE:**

Submit your solution of worksheet-1[named as “yourid\_task\_1.docx”] along with a text file containing URL of your GIT REPO zipped in file named “YourID.zip” in the link provided on the VLE of UWE.

**Tasks**

**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 x 9/5) + 32

C = (F - 32) x 5/9 **[10 marks]**

Answer:

#include <iostream>

using namespace std;

class TemperatureConverter {

private:

int choice;

double temp;

double op;

public:

void showMenu() {

cout<<"Press 1 for celsius to fahrenheit!!!"<<endl;

cout<<"Press 2 for fahrenheit to celsius!!!"<<endl;

getchoice();

}

void getchoice(){

cout<<"enter your choice:"<<endl;

cin>>choice;

getTemperature();

}

void getTemperature() {

if (choice==1){

getinput();

convertToFahrenheit();

}

if (choice==2){

getinput();

converttocelsius();

}

else{

cout<<"Invalid Response";

}

}

void getinput(){

cout << "Enter temperature ";

cin >> temp;

}

void converttocelsius(){

op = (temp - 32) \* 5/9;

displayResult();

}

void convertToFahrenheit() {

op = (temp \* 9.0 / 5.0) + 32;

displayResult();

}

void displayResult() {

cout << "The conversion of temp. is: " <<op<< endl;

}

};

int main() {

TemperatureConverter converter;

converter.showMenu();

return 0;

}

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.

**[10 marks]**

Answer:

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

class NumberGuessingGame {

private:

int maxNo;

int randomNo;

int attempts;

public:

NumberGuessingGame(int max) : maxNo(max), attempts(0) {

srand(time(0));

randomNo = rand() % maxNo + 1;

}

void play() {

int guess;

cout << "Guess the number between 1 and " << maxNo << ": ";

do {

cin >> guess;

attempts++;

if(guess < randomNo)

cout << "lower! Try again: ";

else if(guess > randomNo)

cout << "higher! Try again: ";

} while(guess != randomNo);

cout << "Congratulations! You guessed the number " << randomNo << " correctly in " << attempts << " attempts.\n";

}

};

int main() {

int choice;

cout << "Select difficulty level:\n";

cout << "1. Easy (1-8)\n";

cout << "2. Medium (1-30)\n";

cout << "3. Hard (1-50)\n";

cout << "Enter your choice: ";

cin >> choice;

int maxNo;

switch(choice) {

case 1:

maxNo = 8;

break;

case 2:

maxNo = 30;

break;

case 3:

maxNo = 50;

break;

default:

cout << "Invalid choice. Exiting program.\n";

return 1;

}

NumberGuessingGame game(maxNo);

game.play();

return 0;

}

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

**[10 marks]**

Answer:

#include <iostream>

using namespace std;

class Array {

public:

Array(int size) : size(size), arr(new int[size]) {}

~Array() {

delete[] arr;

}

void readArray() {

cout << "Enter " << size << " integer numbers: ";

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

cin >> arr[i];

}

}

void findMinMax() {

int min = arr[0];

int max = arr[0];

for (int i = 1; i < size; i++) {

if (arr[i] < min) {

min = arr[i];

}

if (arr[i] > max) {

max = arr[i];

}

}

cout << "Minimum value: " << min << endl;

cout << "Maximum value: " << max << endl;

}

private:

int size;

int\* arr;

};

int main()

{

int size;

cout << "Enter the size of the array: ";

cin >> size;

Array a1(size);

a1.readArray();

a1.findMinMax();

return 0;

}

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. **[20 marks]**

Ans:

#include <iostream>

using namespace std;

int main() {

int dayNumber;

cout << "Enter a number (1-7) to get the corresponding day of the week (1 = Sunday): ";

cin >> dayNumber;

// Check if input is valid

if (cin.fail()) {

cout << "Invalid input. Please enter a valid number." << endl;

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

}

switch(dayNumber) {

case 1:

cout << "The day is: Sunday" << endl;

break;

case 2:

cout << "The day is: Monday" << endl;

break;

case 3:

cout << "The day is: Tuesday" << endl;

break;

case 4:

cout << "The day is: Wednesday" << endl;

break;

case 5:

cout << "The day is: Thursday" << endl;

break;

case 6:

cout << "The day is: Friday" << endl;

break;

case 7:

cout << "The day is: Saturday" << endl;

break;

default:

cout << "Invalid input. Please enter a number between 1 and 7." << endl;

}

return 0;

}

**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]**

**Answer:**

#include <iostream>

using namespace std;

bool isBouncy(int number) {

bool increasing = false;

bool decreasing = false;

int lastDigit = number % 10;

number /= 10;

while (number > 0) {

int currentDigit = number % 10;

if (currentDigit < lastDigit) {

increasing = true;

} else if (currentDigit > lastDigit) {

decreasing = true;

}

// If both increasing and decreasing, it's bouncy

if (increasing && decreasing) {

return true;

}

lastDigit = currentDigit;

number /= 10;

}

return false;

}

int main() {

int num;

cout << "Enter a positive integer: ";

cin >> num;

if (cin.fail() || num <= 0) {

cout << "Invalid input. Please enter a positive integer." << endl;

return 1;

}

if (isBouncy(num)) {

cout << num << " is a bouncy number." << endl;

} else {

cout << num << " is NOT a bouncy number." << endl;

}

return 0;

}

**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:
   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

Answer:

#include <iostream>

using namespace std;

const int ROWS = 5;

const int COLS = 5;

void displaySeats(char seats[ROWS][COLS]) {

cout << "\nSeating Arrangement:\n";

cout << " ";

for (int c = 0; c < COLS; ++c)

cout << c + 1 << " ";

cout << endl;

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

cout << i + 1 << " ";

for (int j = 0; j < COLS; ++j) {

cout << seats[i][j] << " ";

}

cout << endl;

}

}

int main() {

char seats[ROWS][COLS];

// Initialize all seats as available

for (int i = 0; i < ROWS; ++i)

for (int j = 0; j < COLS; ++j)

seats[i][j] = 'O';

int row, col;

char choice;

do {

displaySeats(seats);

cout << "\nEnter row (1-5): ";

cin >> row;

cout << "Enter column (1-5): ";

cin >> col;

// Input validation

if (row < 1 || row > 5 || col < 1 || col > 5) {

cout << "Error: Invalid row or column number. Please enter values between 1 and 5.\n";

}

else if (seats[row - 1][col - 1] == 'X') {

cout << "Error: Seat already booked. Please choose another seat.\n";

}

else {

seats[row - 1][col - 1] = 'X';

cout << "Seat successfully booked.\n";

}

displaySeats(seats);

cout << "\nDo you want to book another seat? (y/n): ";

cin >> choice;

} while (choice == 'y' || choice == 'Y');

cout << "Thank you for using the cinema booking system!\n";

return 0;

}

**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.