**Labsheet 12 - Programs using Classes**

All students should be working on this labsheet this week. I have put the labsheet template projects on your shared OneDrive. Students should aim to get Q1 from labsheet 12 done this week and Q2 next week at a minimum.

There is a **template project** and **executable** for each of the questions.

**Remember to design your programs first. Name the programs appropriately.**

# **It is absolutely prohibited to copy anyone else’s design or code. You can ask for help with a particular problem from friends, colleagues, the lecturers in the lab etc but you must write your own design and code the fix to the problem yourself.**

**If you get help from someone or take design/code from the web or elsewhere, you have to comment in your code to state what help you got and from where.**

**Saving Your Projects**

You should save the projects for this labsheet into a folder called **labsheet 12** within **Programming\Labsheets** folder on your **M drive.**

Every program should have a **comment** at the top stating your name, student ID number, date created, approx number of hours worked on, overall brief description of the program and any known bugs in it.

**Question 1:**

There is a template project called **Player - template** on the shared OneDrive in the following folder:

**Programming\Labsheets\labsheet 12**

You should download a ***copy*** of this template project to your **M drive.** You will need to uncompress the folder before you can use it after downloading it. You do this by right clicking on the project folder and then choosing ***extract***.

**You should use and modify this template project in order to do this question.**

This project contains the following Player class:

class Player

{

// the data members are private by default

sf::Texture texture; // texture for player

sf::Sprite sprite; // sprite used to represent player

int image\_width; // the width of the image in pixels

bool scaled; // is the image increased in size

int speed; // the speed that the player moves

int lives; // the number of lives

int direction; // is the player moving east or west

};

The player has an image for a body. You will need to add ***new functions*** to this class. For example you will need to add public data functions to access the private data members where required in the Player class.

There is also a **Game** class which contains a **player object** called **PlayerOne** which isa **data member** as follows:

class Game

{

// the data members are private by default

Player playerOne; // create an object of type player

public: // declaration of member functions

void loadContent();

void run();

};

You **DO NOT** need to make any changes to the Game header file.

The main( ) function within the Game.cpp file looks as follows:

int main()

{

**Game myGame;**

myGame.loadContent(); // load the font file

**myGame.run();**

system("Pause");

return 0;

}

An instance (object) of the Game class called ***myGame*** is created in the main( ) function. The Game object calls the run( ) function which contains the ***main game loop***. The player object can be initialized and used as required for the game within the run( ) function. For information on a game loop within a run() function see page 31 of the Chapter 8 notes.

There is also a **Globals.h** file. You **do not** need to make any changes to this file. All the global variables for the project are contained within this file.

**You should add the following functionality to this program:**

1. The player should have the following movement: It should move from left to right across the screen. When it reaches the boundary of the screen on the right hand side it should move from right to left. When it reaches the boundary on the left hand side it should move from left to right and so on.
2. When the user presses the ‘f’ key on the keyboard, the player’s movement speed should increase by 1. The player’s speed should not increase more than 15.
3. When the user presses the ‘s’ key on the keyboard, the player’s movement speed should decrease by 1. The minimum player’s speed is 1.
4. When the user presses the ‘i’ key on the keyboard, the player’s size should increase. This can be done by scaling the image. This should be done ONLY once. Your code should prevent it happening more than once.
5. When the user presses the ‘c’ key on the keyboard, the player’s colour should change.
6. When the user presses the ‘d’ key on the keyboard, the player should move in the opposite direction.
7. Add a second player object to the project called playerTwo. You will need to change the Game.h file to do this. Get your playerTwo object to move like playerOne. It should also increase and decrease in speed when the keys are pressed. It should behave like playerOne.

**Important:**

There is also an **executable** of what the program should do on the shared oneDrive within the same folder. You should run this executable. It is called **PlayerProject.exe**

**Question 2:**

You are given a **Dice** class that represents a 6 sided dice in a game. The Dice class should implement the following functions:

Dice(); // default constructor

void rollDice(); // generates a new number randomly for the dice

int getDiceNum(); // returns the dice number only

You have been given a **template** project called “Dice Game - template”. You should download a copy of this project folder onto your M drive. Modify the Dice and Game code files in this project. You **DO NOT** need to make any changes to the Game header file.

An object of type Dice is a data member of the Game class.

Dice gameDice; // variable to store the new Dice object

Modify the project to ***randomly*** generate another dice number when the up key is pressed. The dice number should be displayed on the screen.

The template project is on your shared OneDrive in the following folder:

**Programming\Labsheets\labsheet 12**

There is also an **executable** of what the program should on your shared OneDrive within the same folder. You should run this executable. It is called **Dice Game.**

**Question 3:**

You have been given a **CardDeck** class that represents a deck of cards in a game.

class CardDeck

{

// the data members are private by default

std::string cardSuit; // the suit of the card eg Hearts, Spades etc

std::string cardFace; // the number of the card eg Ace, 3, Queen etc

public: // declaration of member functions

CardDeck(); // default constructor

void dealCard(); // Randomly generate a new card

std::string getCardSuit(); // function should ONLY return the cardSuit

std::string getCardFace(); // function should ONLY return the cardFace

};

The function **dealCard( )** should randomly generate a new card. It should randomly generate a number between 1-4 for the **card suit.**

If a 1 is generated the cardSuit should be "Hearts" and so on.

It should randomly generate a number between 1-13 for the **card face**.

If a 1 is generated the cardFace should be "Ace", if a 2 is generated the

cardFace should be "2" ... if 11 is generated the cardFace should be "Jack" if 12 is generated the cardFace should be "Queen" and so on.

It should not return anything.

You have been given a template project called “**CardDeck - template**”. You should download a copy of this project folder onto your M drive. Modify the CardDeck and Game code files in this project. You **DO NOT** need to make any changes to the header files.

An object of type CardDeck is a data member of the Game class.

CardDeck aCardDeck; // variable to store the new card desk object

Modify the project to ***randomly*** generate a new card when the down key is pressed. The number (face) of the card and suit of card should be displayed on the screen.

The template project is on your shared OneDrive in the following folder:

**Programming\Labsheets\labsheet 12**

There is also an **executable** of what the program should do on your shared OneDrive within the same folder. You should run this executable. It is called **CardDeck.**

**Question 4:**

Copy the template project **Q4 - Using Menu Buttons** to your **M** drive.

This template project contains code to run a 3 button menu. You can click on the menu with your left mouse and it calls the game to play or the instructions to display or to exit.

A picture containing text, device, scoreboard, meter

Description automatically generated

Have a look at the **MainMenu** class and see if you can understand how this class is designed and how it works with the **Game** class.

How would you add the **Player** class designed in Q1 to this project? You should modify the project so that you add an object of type player to the Game class and the player should display and move, increase in size etc when the Play button is clicked.

**Note** the use of the **gameMode** data member in the **Game** class. It is a good idea to use game modes in your project as it allows you to have a better structure to your project. The gameMode variable is used to represent the different states a game can be in. For example: in Instructions mode (user is reading the instructions) or in Game play mode (user is playing the game) or User input mode (user is inputting some game data like their name).

When you press the Escape key, you return from the game play mode and the instruction mode to the menu mode.

See notes on the use of gameModes in Chapter 10 Games Modes in C++ in your Notes folder on your OneDrive.