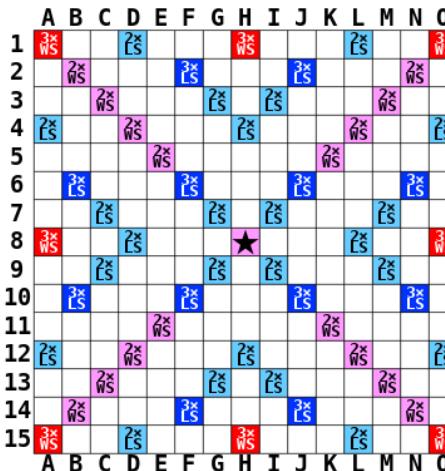
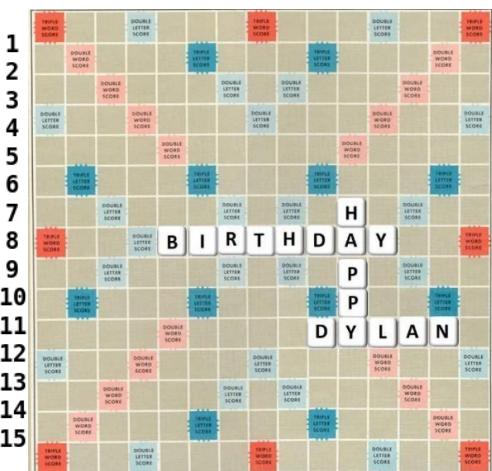


Structured Programming 2020-2021

Projects Description

1.	My Pet Schedule
Description	This program is developed to allow the user to keep the data of the pets he/she owns.
Deliverables	<ul style="list-style-type: none">• User can login (username, password)• Multiple user can use the program (store users data) <p>After logging in, a list of options should appear that contains:</p> <ul style="list-style-type: none">• add a new pet (name, type, date of birth, list of every treatment it takes and the date each treatment, the date of the upcoming treatment)• update any data of an existing pet• delete a pet• display all the pets (all data + age)• display all the pets ordered by the date of the upcoming treatment (all data + age)• search for pets by the date of the upcoming treatment (all data + age)• The program must continuously loop until the user chooses to close the application.
Must be done	<ul style="list-style-type: none">• Use arrays, structs and functions• Use C++• Use files to store data• Apply the concepts you study through the course
Bonus	<ul style="list-style-type: none">• GUI

2.	HangMan
Description	This project aims to develop a game that lets the player guess a word. The player has a certain number of trials according to the game level.
Deliverables	<ul style="list-style-type: none"> o You should store several words representing several categories (at least 4). o You must have (scientists, programming) in the categories. o Each category must have at least 50 words o Words must have different lengths to be used in different levels o The game should have 3 levels (easy, medium, hard) o Each level has 10 trails (if the player correctly guesses 10 words then he passes the level) o Easy level: words from 5 to 7 characters, 7 wrong guesses o Medium level: words from 7 to 9 characters, 6 wrong guesses o Hard level: words from 9 to more than 9 characters, 5 wrong guesses o The player will be able to choose the levels and the categories. o If the player guesses the word correctly before the trails end, then he wins and automatically the second word will appear o If the player runs out of trails then he loses, and the correct answer will appear then another guessing will appear to continue the game. o If the player wins the 10 guesses of the level, then automatically he will pass to the upcoming level. o After finishing the 3 levels, ask the player if he wants to continue playing. Then start again and let him choose the level and the category. Then let him guess random words from your stored words. (Don't just repeat the previous words)
Must be done	<ul style="list-style-type: none"> • Use arrays, structs and functions • Use files to store data • Use C++ • Apply the concepts you study through the course
Bonus	<ul style="list-style-type: none"> • GUI • Multiplayer (Save data for players and provide sign in and sign out

3.	Scrabble
Description	Scrabble is a word game where 2-4 players each put a word on a 15x15 cell board, where each cell holds 1 letter. New words must intersect with the words already on the board. Click here for a video tutorial.
Deliverables	<ul style="list-style-type: none"> Assume only 2 players will play Initialize the board. Assume the board has no special tiles (no colored tiles in the below images): <div style="display: flex; justify-content: space-around;">   </div> <ul style="list-style-type: none"> Write the A-O letters on the top of the board and 1-15 numbers on the left of the board, to make it easy for the player to decide their coordinates. There is a tile bag with 100 tiles (find the letter distribution and the different letter point values here). At the beginning of the game, randomly assign 7 tiles from the tile bag and use the letter tiles available to them to build valid English words on the board. The player whose first tile is the smallest letter starts the game first. The first player must place a word at the middle of the board on the star cell. Words may be placed on the board either vertically or horizontally, diagonal word placement is not allowed. For a player to put down a word on the board, they provide 3 pieces of information: <ul style="list-style-type: none"> the word the start coordinates

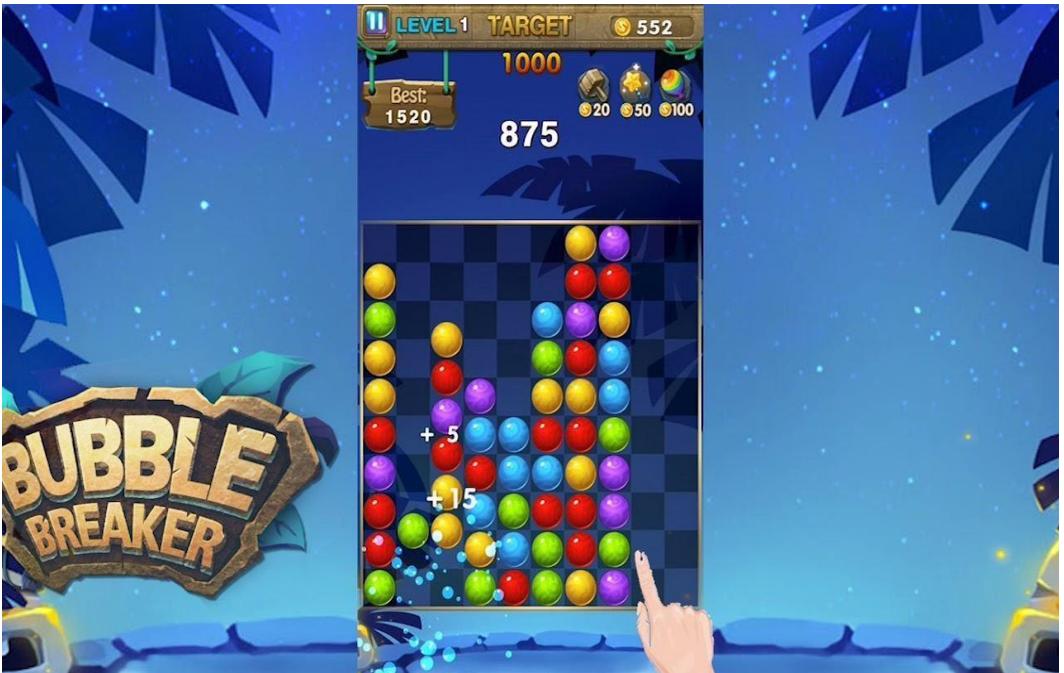
	<ul style="list-style-type: none"> ○ the end coordinates • You must check: <ul style="list-style-type: none"> ○ if the player has enough letters to create this word ○ if the word is a correct word from the dictionary file that is given to you here. You can use larger dictionaries if you want. ○ if the word is not correct (not found in the dictionary file), then show an error message and take a new word. ○ if the word is correct, make sure that it correctly intersects another word on the board (given the input coordinates) – without overriding another word on the board. • If the word is successfully placed on the board (it passed all the previous conditions), then remove the used letters from the player's tiles and calculate their word value. <ul style="list-style-type: none"> ○ Sum the letter point values if the player's word is successfully placed over any special tiles. ○ Record the total value for each player. • Players should not see each other's tiles. Show only the tiles for the current player. When the next player plays, hide the previous player's tiles and show the new player's tiles only. • Players can request new tiles, but at any time they can have a maximum of 7 tiles. • Players can stop and save the game. They can load and continue the game again later. • The first player to finish all their tiles finishes the game. The player with the largest score wins. Show the scores for each player, highlight the winner and show the time taken to complete the game.
Bonus	<ul style="list-style-type: none"> • Create a GUI • Allow for 3 and 4 players • Use the special cells on the board (Double Letter, Triple Letter, Double Word, Triple Word) • Scoreboard

4.	Football Geeks
Description	This project aims to develop a football application that lets the user know the results of past matches and the date and time of upcoming matches for several football clubs .
Deliverables	<ul style="list-style-type: none"> o You must work with several teams (at least 7 teams) o Each team must have at least two past matches stored in the application o You must get the upcoming matches for the teams in your application for the next month . (the real matches with their real date and time) o Admin can update results of match , add new upcoming matches , add new teams o user can follow teams , see results of past matches and see the date and time of upcoming matches
Must be done	<ul style="list-style-type: none"> o Use arrays, structs and functions o Use files to store data o Use C++ o Apply the concepts you study through the course
Bonus	<ul style="list-style-type: none"> o GUI o Provide league table for teams and current standing o The application must have Login functionality (Admin & User) o The application must have (filter by) functionality that show only result of selected team o Make your application update the results of matches live is a huge plus

5.	Travel Agency Management System
Description	This project aims to create a system for a travel agency, where admins can manage trips and users can manage their reservations.
Deliverables	<ul style="list-style-type: none"> • The Admin has a special username and password set beforehand. • Users can register their account, login and logout. • The admin can: <ul style="list-style-type: none"> ◦ add trips. Trips have a trip ID, start date, end date, destination, means of transportation, hotel name, hotel address, type of boarding, maximum number of reservations, number of reservations made, trip cost. ◦ update trips' information. ◦ delete trips. ◦ show all reservations – either for all trips or for a specific trip. ◦ show all users – either all registered users or users with certain reservations. ◦ logout. • When a user makes a reservation, the number of reservations for that trip increases, and the opposite when they cancel a reservation. • The user can: <ul style="list-style-type: none"> ◦ register a personal account. Users have a unique username, first name, last name, phone number, email, address, password. Validate that the username is unique. Validate that the phone number and email are in their correct formats. During registration take a password and a password confirmation (do not store the password confirmation). ◦ edit personal account information. ◦ login to personal account. ◦ reserve a trip. A Reservation includes the number of the people, payment method, confirmed payment or not. If the reservation is successful, show a confirmation message. ◦ change a reservation. If the change is successful, show a confirmation message.

	<ul style="list-style-type: none"> ○ see the history of all their reservations with the details of all the reservations. ○ see all their open reservations with the details of the reservations (trips they did not go on yet). ○ cancel reservation. The user can cancel a reservation only if there is more than 1 week left before the starting trip date. If there is 1 week or less left, then the reservation cannot be canceled – show an error message. ○ request refund or leave their money in their system ‘wallet’ for their next reservation. ○ logout. <ul style="list-style-type: none"> ● You must use structures for at least: Trip, User, Reservation ● You must use files (read and write). ● The program must continuously loop until the user or admin explicitly chooses to close the application. ● Color text as you see suitable (for example, error messages should be red, confirmation messages should be green, etc...).
Bonus	<ul style="list-style-type: none"> ● If you have any bonus ideas, consult your supervising project TA first.

6.	Mini Search Application
Description	<ul style="list-style-type: none"> • This project aims to search on some keywords in a group of documents and return the name of the document that contains this word. • The project also has options to delete and modify the selected word.
Deliverables	<p>The application should:</p> <ul style="list-style-type: none"> • Search for keywords. • Keyword searches should be case insensitive. • Display menu for user to select his choice of operation. • Add Words. • Delete the selected word. • Update or change the selected word. • Saving data in a file or more so that when closing the program and reopening it the data still exists. •
Bonus	<ul style="list-style-type: none"> • Graphical User Interface (GUI). • Search for a given keyword and return the set of documents containing the keyword. • Any extra functions check with project supervisor first.

7.	Bubble Breaker
Description	<p>The aim of this game is to arrange big areas of bubbles of the same color. The more bubbles of the same color are part of an area, the more points you will get.</p> <p>At the beginning the field is initialized and random color values are assigned to the bubbles. The High Score is loaded from a file. By selecting a bubble for the first time, the game calculates the possible points for removing the area around the bubble. Clicking a second time, all bubbles in the area which have the same color are removed. Only areas with more than one bubble are removed.</p> <p>In the next step the program checks if there are areas remaining or if the game is over. When the user finishes the game and reaches a new high score, it is stored in a file. A new game can be started.</p>  <p>click here to watch the game video</p>

Deliverables	<ul style="list-style-type: none"> • The main program, which initializes the game, controls the user action and checks if the game is over. • The selected bubbles are stored and the possible points calculated. • Randomly generates one out of five colors. • This program proves if there is an area with at least two bubbles of the same color If not, the game is over. • Generates a random value, which is used to generate a color for the bubbles. • Based on the selected bubble, the area with bubbles of the same color is marked and the according points are calculated. When the area consists of more than one bubble of the same color all bubbles are removed. • Emerging gaps will be filled with the remaining bubbles and empty rows removed. • Pressing the New Game button initializes a new game • Depending on the score at the end of the game one of three banners is displayed.
Bonus	<ul style="list-style-type: none"> • Saving in a text file the history of the game • in the The High Score displays the value from the High Score file. Score shows the current point. • Scoreboard

8.	Brick Breaker
Description	<p>It has a small ball that hits the bricks with the help of a little platform at the bottom. The player uses this platform to bounce the ball. The more you break the bricks, the more you score. If you miss the ball to bounce then game over</p> 
Deliverables	<ol style="list-style-type: none"> 1. Display bricks and disappear after hitting the ball. 2. Small platform moving left and right 3. The ball must bounce after hitting brick and platform at the bottom
Bonus	<ul style="list-style-type: none"> • Saving in a text file the history of the game • ScoreBoard • Score display

9.	Self-Diagnosis System
Description	Self-Diagnosis System is a system that can help regular people diagnose themselves based on their symptoms that the doctors specified to the system.
Deliverables	<ul style="list-style-type: none"> • Log In (Username & Password). <ul style="list-style-type: none"> ◦ If the username corresponds to a doctor, the following list should be displayed: <ul style="list-style-type: none"> ▪ Add disease ▪ Remove disease ▪ Add symptom to an existing disease ▪ Remove a symptom from an existing disease ▪ Display All diseases. ▪ View Patients with a certain disease ◦ If the username corresponds to a patient, the following list should be displayed: <ul style="list-style-type: none"> ▪ Find out your disease based on your symptoms ▪ Display symptoms for a certain Disease ▪ View diagnosis history ▪ Clear diagnosis history • Register as doctor/patient
Bonus	<ul style="list-style-type: none"> • Edit basic info (username/password, age) • If a patient doesn't find a disease that matches their symptoms 80%. To find the closest disease and print the percentage it matches that disease. <p>Any New Idea, check with the project supervisor first.</p>
Notes	<p><u>Structures should be used :</u></p> <p>Patient: Fullname , account , age , gender , disease_History (array of disease ID).</p> <p>Doctor : Fullname, account, diseases_added (array of disease ID)</p> <p>Disease: Title, ID, general_info (string), symptoms (Array of strings) , no_patients_diagnosed.</p> <p>Account : username, password.</p> <p><u>General Notes :</u></p>

- All doctors have access to view diseases added by different doctors .
- In order for a doctor to change / add a symptom, it must be to a disease they added. otherwise, they don't have access.
- To diagnose a patient with a disease they must display at least 80% of the symptoms of that disease.
 - If no exact disease can be found (with 80% match),the search will display “no exact match found” .
- When a patient is diagnosed, the general info of the disease is shown along with the percentage of patients that were diagnosed with that disease.
- The program should read from an input file, all the diseases/doctors/patients info into an array of structures (Disease,Doctor,Patient - mentioned above) and those arrays will be updated during the run of the program.

10.	Cinema Booking System
Description	This program aims to handle cinema reservations for the available movies in a certain time.
Deliverables	<ul style="list-style-type: none"> • List all available movies in cinema with their description (Title, Director, Actors, Rating) • List theatres showing a certain movie with its screening time. • List all movies showing in a certain time (starting time of the movie) . • Book a ticket. (Check the link in the Notes for more info). When you complete the reservation, display the theatre ID.
Bonus	Any New Idea, check with the project supervisor first.
Notes	<p><u>Structures should be used :</u></p> <p>Movie: Title, Director, Actors, Rating and array of <i>Screening_info</i> (array of structures)</p> <ul style="list-style-type: none"> • Screening_info : Theatre , screening_time(using structure <i>Time</i>). • Theatre : theatre_id, capacity , resevered_seats • Time : hour , minute. <p><u>General Notes :</u></p> <ul style="list-style-type: none"> • To check whether there are available seats for a certain movie , check (capacity - reserved seats) for each theatre screening the movie at a certain time. • Use a global variable with type “<i>time</i>” that can be initialized to simulate the time the program is used (the time the user is booking the ticket/s) . • The program should read from an input file, all movies' info into an array of <i>Movie</i> (structure mentioned above) . • In the input file we will assume that : <ul style="list-style-type: none"> ◦ The duration of all movies is 2 hours. A half an hour is left between each screening in the theatre.

	<ul style="list-style-type: none"> ○ Number of Movies: 4 (at least) , Number of theatres : 2 (at least), Number of screenings in each theatre : 4 (at least). ○ The theatres screen the movies simultaneously. ○ Each movie is screened at least 2 times. <p><u>Booking a Ticket :</u> Click Here .</p> <p><u>Cinema Project FAQ :</u> Click Here</p>
--	--

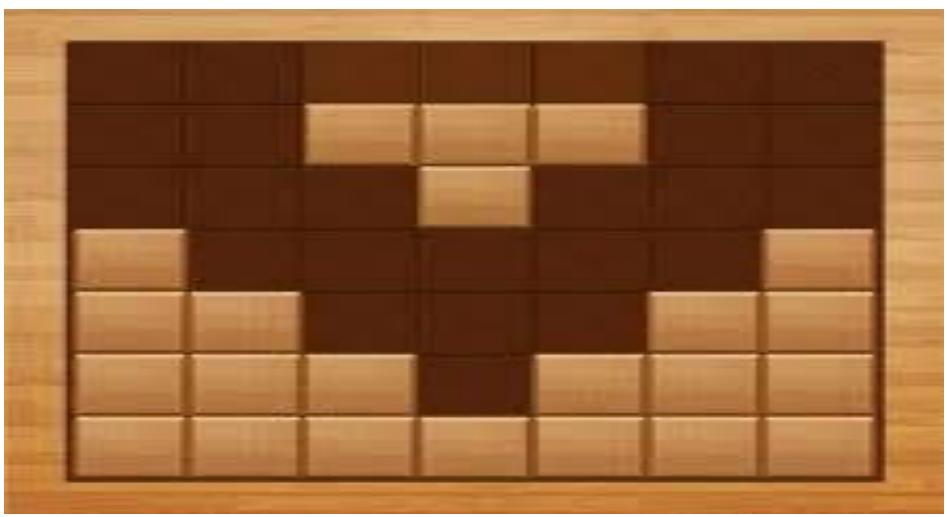
11.	Banking Record System
Description	Implement a banking record system that manages a clients account.
Deliverables	<ul style="list-style-type: none"> • Log In (Username & Password). • If the username corresponds to a admin, the following list should be displayed: <ul style="list-style-type: none"> ○ Add a New Account. ○ Modify An Account. ○ Close An Account. ○ List All Accounts. • If the username corresponds to a client, the following list should be displayed: <ul style="list-style-type: none"> ○ Balance Enquiry. ○ Withdraw Amount. ○ Deposit Amount. ○ Transfer from his account to another account. • Register as admin/client • Log out
Bonus	Any Extra Functions, check with the project supervisor first.

12.	Quiz Game
Description	<ul style="list-style-type: none"> • In this project a number of questions are asked and the user is awarded a cash prize for each correct answer given. • Questions are chosen in such a way that they cover some fields of a typical quiz contest. The user's general knowledge is tested with quiz questions regarding science, technology, movies, sports, general health, and many more.
Deliverables	<p>The application should:</p> <ul style="list-style-type: none"> • Store the user's name • view the highest score of the user, and even reset the score. • Divides into two rounds. Users must pass the first round to reach the second one • Displaying all question fields. • The user chooses the field. • Saving the users' name and their highest score in a file or more so that when closing the program and reopening it the data still exists.
Bonus	Any Extra Functions, check with the project supervisor first.

13.	AqarMap
Description	Aqarmap services allow you to buy or sell an apartment while providing essential information (i.e. price, location, area,etc.) to help you take the right decision.
Deliverables	<ul style="list-style-type: none"> • <u>System Data (Structs should be created for) :</u> <p>Apartment Owner: Owner ID, Name, Phone, E-mail, List_of_apartments</p> <p>Apartment Info: Apartment ID, Location (using location struct), no_of_rooms, FloorNo., Area, Price, View, PaymentMethod, DateOfAdvertisement</p> <p>Location: StreetName, City</p> <p>Apartment Installments (تقسيط): Apartment ID, Down Payment (مقدم), Installements_Years</p> • <u>System Functionalities :</u> <p>Owner Functions:</p> <ol style="list-style-type: none"> 1. Sign up 2. Log in 3. Advertise for an apartment (Offer an apartment for sale) 4. Edit an Advertisement 5. Remove an Advertisement 6. Display All his own Advertisements <p>Client Functions:</p> <p>Search for an apartment according to different filters as follows:</p> <ol style="list-style-type: none"> 1. search by location 2. Search with certain Price Range 3. Search by payment method 4. Search by area 5. search by view 6. search by no. of rooms

	Note that a client could search using combined filters (For example: search by location and search by area and search by payment method)
Notes	<ol style="list-style-type: none"> 1. Information for Apartment Installments will be available only for apartments that offer payment by installments. 2. Example for Apartment view: Main Street, Side Street, Garden, Pool, Nile, etc. 3. Payment method could be either Cash or Installment(تقسيط)
Bonus	<p>advertise for different property types (i.e. villas, Land, Building, Roof, etc.)</p> <p>Any other feasible functionalities approved by your Mentor TA</p>

14.	BookShop
Description	An electronic bookshop has the capabilities of storing and retrieving information about books, customers, and orders.
Deliverables	<ul style="list-style-type: none"> • <u>System Data (Structs should be created for) :</u> <p>Shop Admin: userName, Password Author: Author ID, Name, Date_of_Birth Book: ID, Title, Category , List_of_Authors, Publication_Year, Price, Quantity_in_Stock Customer: UserName, Password, Email, Address, Phone Order: Order ID, Customer ID, OrderDate, List_of_Book_IDs, Total Price, Ship Date</p> • <u>System Functionalities :</u> <p>Shop Admin Functions:</p> <ol style="list-style-type: none"> 1. Log in 2. Add New Author 3. Add New Book for existing author 4. Edit Book Info 5. Delete Author/Book <p>Customer Functions:</p> <ol style="list-style-type: none"> 1. Sign up 2. Log in 3. Search for a book by title and/or by author name and/ or by category 4. Add new order 5. Edit Order 6. Cancel Order 7. Show Order Receipt
Notes	When a customer adds a new order for a certain book, the quantity_in_stock for this book should be decreased, and vice versa when he cancelled an order.
Bonus	Admin could search for orders not yet shipped. Admin could display monthly, quarterly or annual Reports. Any other feasible functionalities approved by your Mentor TA

15.	block puzzle
Description	<ul style="list-style-type: none"> . Drag the blocks to fill the grid with one column or one row . When you create a full line on the grid, you will get scores. . The Blocks can be rotated . Game will be over if there is no space for any of the pending shapes. <p>https://youtu.be/im7pAkqByJI</p> 
Deliverables	<p>Implementing a description with :</p> <ol style="list-style-type: none"> 1- 10 columns , 10 rows. 2- different shapes such as rectangle and square 3- simple GUI
Must be done	<ul style="list-style-type: none"> • Use arrays, structs and functions • Use C++ • Apply the concepts you study through the course
Bonus	<ol style="list-style-type: none"> 1- advanced GUI 2- Sound 3- keep the highest score. 4- more functionality

16.	xonix
Description	<p>The goal is to get the red block out of a six-by-six grid full of wooden blocks by moving the other blocks out of its way.</p> <p>Solve each stage without any hints to earn 3 stars and a super crown!</p> <ul style="list-style-type: none"> • Move the red block to the exit. • Horizontal blocks can move from side to side. • Vertical blocks can move up and down. <p>https://youtu.be/zJCbKPNEnF8</p> 
Deliverables	Implementing a description with : 1- GUI 1- keep the highest score
Must be done	<ul style="list-style-type: none"> • Use arrays, structs and functions • Use C++ • Apply the concepts you study through the course
Bonus	1- more levels 2- Sound 3- save progress of the player (The game should contain levels. Each level should be opened only if the player had passed the previous levels.and the player can be begin from any level as long as the level had been opened before) 4- more functionality

17.	Icy Tower
Description	<p>The player starts the game on the tower's ground floor. To climb the tower, the player must jump from floor to floor. The character can pass through the floors from below but lands on them when falling from above. Once the player has landed on a floor, he can move across its surface using the arrow keys.</p> <p>https://youtu.be/G7dRrW0mMvI</p> 
Deliverables	<p>Implementing a description with :</p> <ol style="list-style-type: none"> 1- simple GUI 2- save the highest score by(min time of max floor).
Must be done	<ul style="list-style-type: none"> • Use arrays, structs and functions • Use C++ • Apply the concepts you study through the course
Bonus	<ol style="list-style-type: none"> 1- Sound 2- more than one level. 3-save progress of player(The game should contain levels. Each level should be opened only if the player had passed the previous levels.and the player can be begin from any level as long as the level had been opened before) 4- advanced GUI 5- more functionality

18.	sonic
Description	<p>Sonic is a blue anthropomorphic hedgehog who can run at supersonic speeds and curl into a ball, primarily to attack enemies. In most games, Sonic must race through levels, collecting power-up rings and avoiding obstacles and enemies.</p> <p>https://youtu.be/xdc0M_ajlIA</p> 
Deliverables	<p>Implementing a description with :</p> <ol style="list-style-type: none"> 1- simple GUI 2- save the highest score.
Must be done	<ul style="list-style-type: none"> • Use arrays, structs and functions • Use C++ • Apply the concepts you study through the course
Bonus	<ol style="list-style-type: none"> 1- Sound 2- more than one level 3-save progress of player (The game should contain levels. Each level should be opened only if the player had passed the previous levels.and the player can be begin from any level as long as the level had been opened before) 4- add monsters. 5- more functionality.

19.	super mario
Description	<p>The player runs and jumps across platforms and stops enemies jumps to gather coins and reach the goal.</p> <p>Mario should be able to break the stones which may supply him with a multitude of power-ups and items that give Mario special powers such as fireball-throwing and size-changing into both giant and miniature sizes.</p> <p>https://youtu.be/G8yorR_e5kY</p> 
Deliverables	<p>Implementing a description with :</p> <ol style="list-style-type: none"> 1- simple GUI 2- save the highest score.
Must be done	<ul style="list-style-type: none"> • Use arrays, structs and functions • Use C++ • Apply the concepts you study through the course
Bonus	<ol style="list-style-type: none"> 1- Sound 2- more than one level 3-save progress of player (The game should contain levels. Each level should be opened only if the player had passed the previous levels.and the player can be begin from any level as long as the level had been opened before) 4- advanced GUI. 5- more functionality.

20.	Sudoku
Description	<p>a puzzle in which missing numbers are to be filled into a 9 by 9 grid of squares which are subdivided into 3 by 3 boxes so that every row, every column, and every box contains the numbers 1 through 9.</p> <p>https://youtu.be/kvU9_MVAiE0</p>
	
Deliverables	<p>Implementing a description with :</p> <ol style="list-style-type: none"> 1- 9 *9 sudoku 2- final check 3- there saved sudoku at least
Must be done	<ul style="list-style-type: none"> • Use arrays, structs and functions • Use C++ • Apply the concepts you study through the course
Bonus	<ol style="list-style-type: none"> 1- Sound 2- more than one level Difficulty level 3- save the highest score by min time for solving the game. 4- generate different sudoku each time 5- save game. 6- more functionality

21.	Flappy Bird
Description	<p><i>Flappy Bird</i> is an <u>arcade</u>-style game in which the player controls the bird Faby, which moves persistently to the right. The player is tasked with navigating Faby through pairs of pipes that have equally sized gaps placed at random heights. Faby automatically descends and only ascends when the player taps the <u>touchscreen</u>. Each successful pass through a pair of pipes awards the player one point. Colliding with a pipe or the ground ends the gameplay. During the <u>game over</u> screen</p> <p>https://www.youtube.com/watch?v=fQoJZuBwrkUv</p> 
Deliverables	<p>Implementing a description with :</p> <ol style="list-style-type: none"> 1- simple GUI 2- save the highest score.
Must be done	<ul style="list-style-type: none"> • Use arrays, structs and functions • Use C++ • Apply the concepts you study through the course
Bonus	<ol style="list-style-type: none"> 1- Sound 2- more than one level Difficulty level 3- can stop and continue the game. 4- more functionality

22.	Course Management System
Description	The system aims to manage courses and students in a school. Data about course shall be (ID, Name, Instructor Name, #Enrolled Students). Data about a student shall be (ID, Name). Data shared between the course and student shall be (Course Id, Student Id, Grade)
Deliverables	<p>System Functionalities:</p> <ul style="list-style-type: none"> 1-Enroll new student to a specific course. 2- Edit, delete, and display the student information. 3- Add, Edit, Display and delete course. 4- Search for a course by ID then display all students enrolled in this course all with his/her grade in the course. 5- Search for a student by ID then displays the courses he is enrolled in if they exist. 6- Graphical User Interface.
Bonus	Any extra functionality.
Notes	<ul style="list-style-type: none"> 1- Arrays and Structures must be used. 2- Data of courses and students should be stored in two separate files (if additional files required, make them). 3- Data of students shall be loaded before searching for a student. 4- Stored Courses shall be loaded at the beginning of the program. 5- A list of predefined courses shall be shown to user when enrolling a new Student

23.	Budget Tracker
Description	Write an application that keeps track of a household's budget. The user can add expenses, income, and recurring costs to find out how much they are saving or losing over a period of time.
Deliverables	<p><< Minimum requirements of the projects >></p> <ol style="list-style-type: none"> 1. The User can add new expenses 2. The user can add new income 3. The system must have recurring costs (car fuel, house rent,etc...) 4. The user can add recurring cost. 5. List all recurring costs. 6. List all income sources. 7. The user can edit any values "income, recurring cost, etc.." 8. The user can delete any value. 9. Display to the user a quick view on the current status 10. Display a detailed status 11. allow the user to specify a date range and see the net flow of money in and out of the house budget for that time period. 12. All these data must be saved in file
Bonus	Any extra functionality. GUI
Notes	Arrays and Structures must be used.

24.	Basic Calculator
Description	The Project aims to create a calculator that will help scientist to do their calculations
Deliverables	<ol style="list-style-type: none"> 1. User will enter two numbers 2. then User will choose operation to perform on both numbers 3. operations are (Add, Subtract, multiply and power) 4. Your calculator should be supporting unlimited ranges of input numbers
Bonus	Apply mathematical optimization to any of the operations above GUI
Notes	

25.	Phone Book Application
Description	<ul style="list-style-type: none"> • This project aims to develop an "Phonebook Contact "application. • This project is useful to store complete information under a single contact number. • Each contact has personal information such as(Name , Gender , Phone Number , Email) are asked while adding a new contact. • The project also has options to delete and modify the entered contact number and his personal information.
Deliverables	<p>The application Should:</p> <ul style="list-style-type: none"> • Display menu for user to select his choice of operation. • Add Contact. • Show the saved contacts. • Search for contact by (name , phone number ,....) • Delete the added contact. • Update or change the details of contacts. • Saving data in a file or more so that when closing the program and reopening it the data still exists.
Bonus	<ul style="list-style-type: none"> • Graphical User Interface (GUI). • Add multiple information to one contact. EX(one contact has two mail , two phone numbers etc) • Any extra functions.

26.	Airline Reservation System
Description	<ul style="list-style-type: none"> This system is used for the reservations of a particular airline. It incorporates airline schedules, passenger reservations and ticket records
Deliverables	<p>System Data:</p> <ul style="list-style-type: none"> Flight information includes: Flight Number, Number of Available seats, Flight Date, Departure Info (City, Departure time), Arrival Info (City, Arrival time) Passenger information includes: Passenger ID, Name, Address, Phone number Ticket information includes: Ticket Number, Passenger ID, Flight Number, Reservation Date <p>Flight Scheduling Functionalities:</p> <ul style="list-style-type: none"> Add New flight schedule Delete existing flight schedule Update existing flight schedule <p>Passenger Functionalities :</p> <ul style="list-style-type: none"> Log in / sign up Search Available Flights Add new reservation Cancel reservation Modify reservation Show reservation log
Bonus	<ul style="list-style-type: none"> Graphical User Interface (GUI). Any extra functions.

27.	Conference Halls Reservation System
Description	<ul style="list-style-type: none"> The aim is to manage reserving conference halls to do events easily. The System should contain details of each hall (hall name, place, List hall events) so users are able to reserve it in available times. Each User should access his account by name and password where he can check his old reservations and reserve new ones. Every event should be defined by (id, name, start Date, End Date).
Deliverables	<p>The System should:</p> <ol style="list-style-type: none"> 1. Add or delete Hall 2. Each Show Record consists of (show name, start Date, End Date, available seats In each day) where each day has max 100 seat. 3. Show all Shows available in a certain timeline. 4. Show all the details of the selected Hall including List of events. 5. Search the event in which hall 6. Users access their accounts by name and password. 7. Users can see their reserved upcoming Events 8. Users can make reservations at any available hall. 9. Users can cancel the reservation they made. 10. Only reservation is made when the hall has no events at that time. 11. Each Event consists of (id, name, Start Date, End Date). 12. Each User account consist of (name ,password , the reserved events' ID)
Bonus	<ul style="list-style-type: none"> • Graphical User Interface (GUI). • Any extra functions.

28.	News Tracker
Description	This project aims to develop a News application that lets the user know the latest news in several categories like (Social , art , sports)
Deliverables	<ul style="list-style-type: none"> o You must work with several categories of news (at least 3 Categories) o Each category must have at least 5 past news o The application must have Login functionality (admin & user) o Admin can add or remove categories , add or remove news , assign news to specific categories o User can follow/ Unfollow categories , read news
Must be done	<ul style="list-style-type: none"> o Use arrays, structs and functions o Use files to store data o Use C++ o Apply the concepts you study through the course
Bonus	<ul style="list-style-type: none"> o GUI o Provide upcoming big events related to each category o The application must have (filter by) functionality that show only result of selected category o Make your application update the news live is a huge plus