TA: Aya Nasser

aya.naser@cis.asu.edu.eg

Title	Events' Scheduler

Description	This is an application to schedule the events for the user.
	● Each event must have:
	1. A name
	2. A start date
	3. An end date
	4. A place
	5. Event's start time
	6. Reminder time … etc
	7. Done (Yes/No)
	● The user must be able to:
	1. Add,
	2. Delete or
	3. Update an event.
	● The application should refuse to add an event that intersects with another event.
	● The user can display his events sorted by start date and reminder time.
	● When an event is done it should
	   ○ disappear from the upcoming events.
	   ○ be added in another data structure holding only the done events.
	Store your data in files, no need for database

Minimum requirements	● You should use the appropriate Data structure for each requirement (**selection of DataStructure will be graded**). ● Clean and Efficient Code is a must. ● You can Use the built in DataStructures Libraries (eg.STLs in C++). ● You must fulfill **all** the above requirements.
Bonus Opportunities	● Any extra modules (**Non-Trivial**) designed to give more functionality to your application. ● Good GUI.
Development Tool	● C++ ● Visual Studio

Title	Employees Task Manager
Description	The Businessman wishes to get an application that helps his employees to manage their tasks ,enhance the way of managing the tasks by prioritising them & to track the tasks. Implement a C++ program that helps them manage their tasks list so as to be able to follow up what's done & what's not. Use the suitable Data Structure(S) to help you design your system.
Minimum Requirements	**The program should allow the employee to:** 1. Reading the employee's tasks (Task Title, Description, Assigning date, Required ending date, Name of Employee the task assigned to & with priority values (1:10 , meaning 1 least important and most important)). 2. Creating a **Priority List/Queue**. 3. Sort the list descendingly according to priority, so that the most important task is on the top.

	4. Display the **Sorted Priority List/Queue** with all it's details. 5. Update the Queue when a task is done. 6. Reminder with the next task to be done. 7. When all tasks are done, give the employee a good encouraging quote.
Bonus	-Edit the task and/or it's priority. -Give the employee option to postpone a task to another day (Required ending date). - Mobile/Android applications. - Good Looking GUI.
Development Tool	-Visual Studio -C++ Language

TA: Amira Samir

amira_samir@cis.asu.edu.eg

Title	**Blood Bank Management System**
Description	A system for helping patients especially at emergent cases. Blood donation has a great importance in helping to save other people's lives. Implement a C++ program that allows the user to do the following requirements.

Minimum Requirements	● A user could be donor or recipient:
	
	 1. Donor
	   a. Register with her data
	      i. ID
	      ii. Name
	      iii. Mail
	      iv. Password
	      v. Age
	      vi. Gender
	      vii. Blood type
	      viii. If she suffers from any disease (blood pressure disorders, thyroid disease, diabetes, cancer, heart disorders, hepatitis)
	      ix. If she suffers from any other disease or take any medicine
	      x. Date of the latest donation
	   b. Login
	   c. Donation request
	   d. Update her data
	   e. Delete account
	
	 2. Recipient
	   a. Register with her data
	      i. ID
	      ii. Name
	      iii. Mail
	      iv. Password
	      v. Age
	      vi. Gender
	      vii. Blood type
	      viii. Hospital
	      ix. Doctor of the case
	   b. Login
	   c. Update her data
	   d. Delete account

	e. Search for the availability of blood type
	f. Display all blood data
	a. Type
	b. Quantity
	c. Received and Expiry dates
	g. Request the blood type and quantity she wants and confirm about the hospital (place where the patient is available)
	• You should use the appropriate Data structure for each requirement (selection of Data Structure will be graded).
	• Clean and Efficient Code is a must.
	• You can Use the built in Data Structures Libraries (example STLs in C++)
	You should save the data using files **DO NOT** use Database.
Bonus	- The user could be an admin:
	- Validate the donor's request according to her data as a safe donor must be:
	- Age from 17 to 60 years old
	- Last donation date must be at least 3 months before her request
	- She mustn't suffer from any disease from the list
	- Check that she hasn't any other problem including other diseases or medicine
	- If this donor is safe then add her data.
	- Insert/Update/Delete the quantity of blood, and its received and expiry dates
	- Insert/Update/Delete/Display data of donors
	- Insert/Update/Delete/Display data of recipients
	- Any extra (**non-trivial**) features to give more functionality to your application.
	- Amazing GUI.
Development Tool	- Visual Studio (However, you may use any language(s) and IDE(s) you want as long as it implements data structures.)
	- C++ Language

TA: Eman Hamdi

emanhamdi@cis.asu.edu.eg

Title	**Mini** وصلني
Description	Your customer wishes to reach his destination town in the shortest possible time. Implement a C++ program that allows the user to enter the graph which representing the towns then find him the shortest path to reach his destination from any given source town.
Minimum Requirements	-The program should allow the user to: 1. Add graph -towns and distances between them- 2. Update graph data 3. Delete graph 4. Display graph data 5. Determine two towns as source and destination then find the shortest path between them • Compute total distance of the shortest path • Display towns names in the shortest path -You must implement Dijkstra's algorithm to find the shortest path -Implement the graph using Adjacency list
Bonus	Implement more than one algorithm to find the shortest path.

Development Tool	- Visual Studio - C++ Language

TA: Omar Sherif

omarsherif@cis.asu.edu.eg

Title	Chat application
Description	This application helps people to communicate with each other such as the popular application WhatsApp
Minimum requirements	**The application should keep track of these entities.** 1- User : the user has (UserID ,mobile number , password , First name, Last name , ChatRooms, ChatRoomInfo ,Contacts which are other users. , Stories , User Profile description) 2- ChatRoom : the ChatRoom has (ChatRoomID, List of Users (2 in case of one to one chat and 2 or more in case of group chat), List of Messages ,ChatRoom Type (one to one chat or group chat)) 3- ChatRoomInfo : the ChatRoomInfo has (UserID, ChatRoomID , Last seen (Date that corresponds to the last date the user opened the ChatRoom)) 3- Message: The Message entity has (MessageId , UserID (who wrote the message), Text, Status). 4- Status :The status has (Date of message sent , time of message sent , status type (which is either seen or unseen (when the message is sent it is initially unseen until the another user opens the chat room , In case of group chat it is seen if only all the other users opened the chat room)). 5- Story : story entity has (User ID , Published Story time , Story photo (This attribute is optional) , Story text). 6- User Profile Description (User ID, Personal photo if exists , About description, Visibility (Boolean to make your information visible to anyone or only your contacts)) **Constraints:** ● Each user story should be disappeared to all his contacts after 24 hours from the time it was published. ● The user can only see the photo and profile description for the users that already have him as a contact if the visibility is false otherwise he can see any of his contacts photo and profile description. **General functions:**

	- Each user can add new contacts. - Each user can chat with any other user in his contacts and a chatroom should be created for both users. - Each user can create a group chat and a chatroom should be created for all users in the group. - The user can see all the chatrooms that he is already in. - The user can open any of his chatrooms and see all the previous messages and he can also send a new message - The user can see the status of the message whether it is seen or not. - The user can see all the stories of his mutual contacts . - The user can change his profile picture and his about description. - The user can undo a message **Note.** The structure , number , and attributes of the entities aren't necessary to be implemented exactly as mentioned , but the application should achieve all of the constraints and perform all the functions mentioned above .
Bonus	➔ Powerful GUI ➔ Adding any non-trivial function ➔ Applying Best Practices of OOP concepts
Development Tools	- C++ programming language , Any IDE that supports C++

TA: Mohamed Ashraf

mohamed.hassan.std2@cis.asu.edu.eg

Title	Student Prerequisite Subjects Management System
Description	This project requires the use of a suitable data structure to store different courses and their prerequisites and based on this it will allow any student to take any course or not.
Minimum Requirements	- Basic Scenario: This project will take the courses list and their prerequisites list from Admin once then it will allow or disallow the students from taking the prerequisites. - Course Data: 1. Name 2. Code 3. Maximum number of Students. 4. List of pre-required courses. 5. Hours. - Student Data: 1. Name. 2. ID. 3. Password. 4. Finished Courses. 5. Courses in progress.

	6. Academic Year.
	● Student Functionalities:
	1. Log in.
	2. View List of all available courses.
	3. View details of a specific course.
	4. Register for a course.
	5. View all his/her courses.
	6. Edit his/her data.
	● Admin Data:
	1. Name.
	2. Password.
	● Admin Functionalities:
	1. Log in.
	2. Add new student.
	3. Add new course.
	4. Enter course prerequisite.
	5. View List of all students in a specific course.
	6. View List of all courses (Finished - Progressed) of a specific student.
	7. Edit all course data.
Bonus	- *Good Graphical user interface.
	- *Display courses as a graph/tree according to their dependency on one another.
Development Tool	- Visual Studio
	- C++ Language

TA: Mohamed Raafat

MohamedRaafat@cis.asu.edu.eg

Title	**Railway Reservation System**
Description	Railway reservation system automates the process of reserving a railway ticket.
Minimum Requirements	There are two kinds of users:
	1. Admin
	a. Login with the correct password otherwise he is not permitted to use the system
	b. Add/update details of a train
	i. Train name
	ii. Train number
	iii. Boarding point
	iv. Destination point
	v. Number of available seats
	vi. Fare per ticket
	vii. Date of travel
	c. View list of trains
	2. Passenger(s)
	a. Register for the first time [email, password]
	b. Book Ticket with the following information:

	i. Passenger name
	     ii. Boarding point
	   iii. Destination point
	   iv. Date of travel
	   c. The passenger can login to view/update his reservation
	● Your system should keep track of all trains going to the same destination at any time so as to offer for the passenger the time of the next train if the requested one is full.
	● Use a suitable data structures
	● Store data in files or DB
	● Write clean code is a **Must**
	● Apply Object-Oriented programming concepts

Bonus	- GUI (**G**raphical **U**ser **I**nterface). - Applying design patterns. - Applying design principles (ex. SOLID principles).
Development Tool	- Visual Studio IDE, or any IDE that supports C++ - C++ Language

TA: Mohamad Nossier

mohamad.nossier@cis.asu.edu.eg

Title	**Vaccine Tracking System**
Description	"More than 670 million doses of coronavirus vaccine have now been administered across 151 countries. A range of different vaccines is being used to reduce people's chances of getting sick, needing hospital treatment, or dying." [https://www.bbc.com/news/health-51665497] **Implement an "Egyptian Vaccine Tracking System" in C++. That allows the users to enter their personal and vaccination information and let the administrator calculate basic statistics to draw insights from such data.** [**https://www.washingtonpost.com/graphics/2020/health/covid-vaccine-states-distribution-doses/**] What is meant by Egyptian is that it should be specifically tailored to tracking the vaccination process for Egyptians in Egypt in the sense that the National IDs should be 13 numbers, the user can only enter Egyptian governorates, etc To make things easier, we won't account for Egyptians living abroad or non-Egyptian residents. Only Egyptian residents.

Minimum Requirements	The program should allow the user to:
	  1. Users may Add personal record containing:
	       i. Full Name.
	      ii. National ID. [Unique = Shouldn't accept duplicates]
	     iii. Password.
	     iv. Gender.
	      v. Age.
	     vi. Country [Maybe an Egyptian is Living abroad]
	    vii. Governorate.
	   viii. Already vaccinated or applying for vaccination?
	     ix. If vaccinated, received only one or both doses?
	      x. If not, the user should be added to a waiting list.
	  2. Users should be able to display only their own records by entering the National ID and Password.
	  3. Users may Edit/Update any of their information.
	  4. Users may Delete their personal records.
	  5. Can sign in as Admin using Admin password.
	  6. Admin can view or delete (cannot edit/update) all or any record(s) by entering only Nat. ID.
	
	Examples of basic statistics:

	- Percentage of people registered in the system that has received the first dose. - Percentage of people registered in the system that has received both doses. - Percentage of Females and Males registered in the system. - Etc. Evaluation: The evaluation will be mainly based on the students' ability to use and apply the most suitable data structure(s) for the given task(s) and explain why they used these data structures and why they are better than other data structures in any given case. Clean code is a must. Using ready-made libraries with ready-to-use data structures is allowed. However, implementing all the Data Structures yourselves will get you better grades.
Bonus	- Calculate more advanced statistics [Check the Washington post article above] [You may need to ask the user for more information to get more statistical insights]. - Data visualization.
Development Tool	- C++ Language / Visual Studio IDE. (However, you may use any language(s) and IDE(s) you want as long as it implements data structures.)

Title	**Simplified Airbnb**
Description	"After moving to San Francisco in October 2007, roommates and former schoolmates Brian Chesky and Joe Gebbia came up with the idea of putting an air mattress in their living room and turning it into a bed and breakfast. In February 2008, Nathan Blecharczyk, Chesky's former roommate, joined as the Chief Technology Officer and the third co-founder of the new venture, which they named AirBed & Breakfast. They put together a website that offered short-term living quarters and breakfast for those who were unable to book a hotel in the saturated market. The site Airbedandbreakfast.com officially launched on August 11, 2008. The founders had their first customers in town in the summer of 2008, during the Industrial Design Conference held by Industrial Designers Society of America, where travelers had a hard time finding lodging in the city." .. "In March 2009, the name of the company was shortened to Airbnb.com, and the site's content had expanded from air beds and shared spaces to a variety of properties including entire homes and apartments, private rooms, and other properties."[Wikipedia] **Implement a "Simple short-term housing rental" in C++. That allows the "hosts" to enter information about their place, "travelers" to search for a convenient place to rent, and let the system "admin" be able to access, view, and modify all data.** **[https://www.airbnb.com/]**
Minimum Requirements	Both **"Travelers"** and **"Hosts"** may sign-up and provide their personal information (Full Name, E-mail, Password, Nationality, Gender, Age, etc.) + Any other information you think should be provided by the "Traveler" in order to sign-up. After **travelers** log in, they should be able to search for a suitable place to rent given a stay range (start date - end date), stay duration (e.g., 2 days), city, price range, etc., + Any other information you may find relevant to enhance the traveler's search experience. After **hosts** log in, they should be able to add places for rent and provide all the relevant information: availability (start date - end date), city, price, etc., + Any other information you may find relevant to enhance the traveler's search experience. After the **admin** logs in, they should be able to access, view, and edit any data in the whole system. A bonus will be providing an analytics dashboard for the admins (below). **Evaluation:** The evaluation will be mainly based on the students' ability to use and apply the most suitable data structure(s) for the given task(s) and explain why they used these data structures and why they are better than other data structures in any given case. Clean code is a must.

	The requirements are intended not to be very specific, you have the freedom to think and make changes as long as they serve the final product well and you can logically justify such choices/changes.
	Using ready-made libraries with ready-to-use data structures is allowed. However, implementing all the Data Structures yourselves will get you better grades!
Bonus	- Add an analytics dashboard to the "admin" home page showing statistical insights/overview of all the data in the system.
Development Tool	- C++ Language / Visual Studio IDE. (However, you may use any language(s) and IDE(s) you want as long as it implements data structures.)

TA: Manar Sultan

manarSultan@cis.asu.edu.eg

Title	**Modified** صراحه

Description	Modified-Saraha is a website that allows you to send messages in an anonymous way, you don't know any data about the sender <u>except</u> his ID.
	
	● Main class <u>**user**</u> has:
	  ○ ID
	  ○ username
	  ○ password
	  ○ List of Contacts
	  ○ Messages
	- you can add additional helper classes.

Minimum Requirements	- The website should allow the user to: 1. register/login 2. Add a user in his contacts. 3. send a message to a contact. 4. undo the last sent message. 5. search about contact in my contacts (report NOT FOUND if he doesn't exist). 6. view all contacts of specific user. 7. view all the sent messages from latest to oldest. 8. view all the received messages from specific contact. 9. put a message in favorites. 10. remove the oldest message from favorites. 11. view all favorites messages . - Use suitable data structures for requirements (selection of Data Structure will be graded). - Clean and Efficient Code is a must. - You can Use the built in Data Structures Libraries (STLs in C++) - save the data using <u>files</u>
Bonus	- GUI. - any additional useful feature.
Development Tool	- Visual Studio - C++ Language

Title	Mini Text Editor
Description	Develop an application that implements a mini Text Editor. Your application should have the following functions: - Add line: takes text input from user and it will be added to the end of the file - Insert line: takes line number & text input from user and it will be added to specific line. - Get line text: takes line number and return line text - Delete line: takes line number. - Update line: takes line number & text input from user and it will be updated with specific old line. - Find all: takes a string and return all lines numbers which contains that string. - Find and replace all: takes 2 strings then replace first with second in all lines. - Show: Display all lines with their numbers to the user.
Minimum Requirements	- Apply all functions mentioned in the description section. - Insert line & show that complexity of the code is less than θ(N), where N is number of lines. - The project will be a console application. - **Use suitable Data Structures** in your project. (You can use STL)
Bonus	- Any extra (**non-trivial**) features to give more functionality to your application. - GUI.
Development Tool	- Visual Studio - C++ Language

TA: Walaa Alkady

walaa.samir@cis.asu.edu.eg

Title	Online Marketplace Management System

Description	An online marketplace is an e-commerce platform that links sellers with customers.
	- The **seller** can offer **products**.
	- To facilitate the search process, the products are divided into **categories**.
	- Every **customer** can browse or add products to his cart.
	- Display the total cost (receipt) for the customer after his confirmation.
	
	Implement a C++ program that allows the user to choose either he wants to sell or buy a product and then complete the process as a seller or customer using the following:
	- **Seller:**
	    1. ID
	    2. Name
	    3. Email
	- **Product**
	    1. ID
	    2. Name
	    3. Price

	4. Category
	5. Quantity
	6. Seller ID
	- **Customer**
	1. ID
	2. Name
	3. Address
	4. Phone Number
	5. Email
	6. Customer Cart
	- **Cart**
	1. List of Products
	2. Total Price

Minimum Requirements	The program should allow the user to:
	 1. Choose to sell or buy products **(The user can switch between them In The Same Run - You should handle the adding of more than seller and customer)**.
	 2. Seller:
	   - Adding New Seller (Registration): Enter seller information (name, email, …)
	   - Seller login using Email.
	   - Add product and determine its category and quantity .
	
	 3. Customer:
	   - Adding New Customer (Registration): Enter customer information (name, email, …)
	   - Customer login using Email.
	   - Browse products by its category.
	   - Browse products by its name.
	   - Add product to his cart. (Keep track of the available quantity, if the desired quantity is less than the available quantity the customer cannot add this product to his cart).
	   - Confirm the buying and display the total price of the ordered products.
	 Store your data in files or suitable data structures, no need for database

Bonus	- Graphical User Interface (GUI)
	
	 - Add rating system:
	   ● The customers can give an **overview(rate)** out of 5 to the products or the sellers. (Add rate member to the seller and product classes)
	   ● The rate for each product or seller is calculated by the average of all the customers rates.
	   ● Displaying the products to customers according to its rates.

Development Tool	- Visual Studio - C++ Language
Reference Material	Amazon

TA: Mohammed Hosam

mohamed.hossam@cis.asu.edu.eg

Title	**News Management System**

Description	News is important for a number of reasons within a society. Mainly to inform the public about events that are around them and may affect them.
	
	Implement C++ program that allows users to get latest news , filter news by category , Rate news and display them.
	
	- system validates user login data either correct or not.
	
	- system determines the user either admin or not.
	
	- System has many users who can rate and view news.
	
	-System ignores trivial or incorrect news depending on news ratings (below 2) hide from users.
	
	- each news has only one rate for each user.
	
	- news is displayed descendingly according to rates
	
	News :
	    1- Title
	    2- Description
	    3- Date
	    4- Rate (1 to 5)
	
	Admin :
	
	1- Log in
	
	2- Post news to categories
	
	3- Delete news from specific category
	
	4- Update existing news.
	
	User :
	    1- Register by Username and password.
	    2- Log in
	    3- Display latest news
	    4- Show news depending on specific category
	    5- Rate News

Minimum Requirements	- Apply all functions mentioned in the description section. - Store your data in files, no need for database - The project will be a console application. - Use suitable Data Structures in your project. (You can use STL)

Bonus	- GUI. - Add user comments to news where each user can see other comments on news. spam: - Add spam option for each news , if the user reports the post as spam , the post will disappear from this user's news and this news appears to others users with count of how many people spam it .
Development Tool	- Visual Studio - C++ Language

TA: Engy Abdallah

engy.abdallah@cis.asu.edu.eg

Title	**Car Showrooms & Repair Shops**
Description	An application that shows customers Cars' Showrooms to buy/rent a car and Repair shops (Garages) to repair their own cars Implement a C++ program that allows the user to do the following requirements
Minimum requirements	- Admin a) ID b) Username c) Password -Customer a) ID b) Username c) Password d) Buy, Rent or Repair -Showrooms a) ID b) Name c) Location d) Phone number e) List of Cars -Car a) ID b) Make c) Model d) Year e) Price

	f) Installment -Buy/Rent Process a) Process ID b) Customer ID c) Date d) Amount of money e) Chosen car -Garages a) ID b) Name c) Location d) Phone number e) List of Services -Service a) ID b) Name c) Price -Service Process a) Process ID b) Customer ID c) Date d) Amount of money e) Chosen Service - Admin can add, update & delete data related to showrooms, garages, cars & services - Customer can register to the app - Customer can search for specific car, service, showroom or garage - Customer can view list of data related to showrooms/ Garages - Customer can buy/ rent a car or book garage appointment using the app
Bonus	➔ Customer can reserve his picked car, so that it's not shown in the list of available cars of a Showroom for limited time ➔ Show historical services data for one customer ➔ GUI

Development Tools	- Use suitable Data Structures in your project (You can use STL) - Visual Studio
	  C++ Language
	  Save the data using Files