

4IT Erasmus+Project 2024-1-RO01-KA121-VET-000214264
Beneficiary Colegiul National Gheorghe Vranceanu, Bacau, Romania

Sending organization: Colegiul National ”Roman-Vodă” Roman, Romania

MOBILITY PORTFOLIO
Paphos, Cyprus

Mobility period:
PAPHOS, CYPRUS, 13.04.2025-27.04.2025 – practice period 14.04.2025-25.04.2025

Receiving

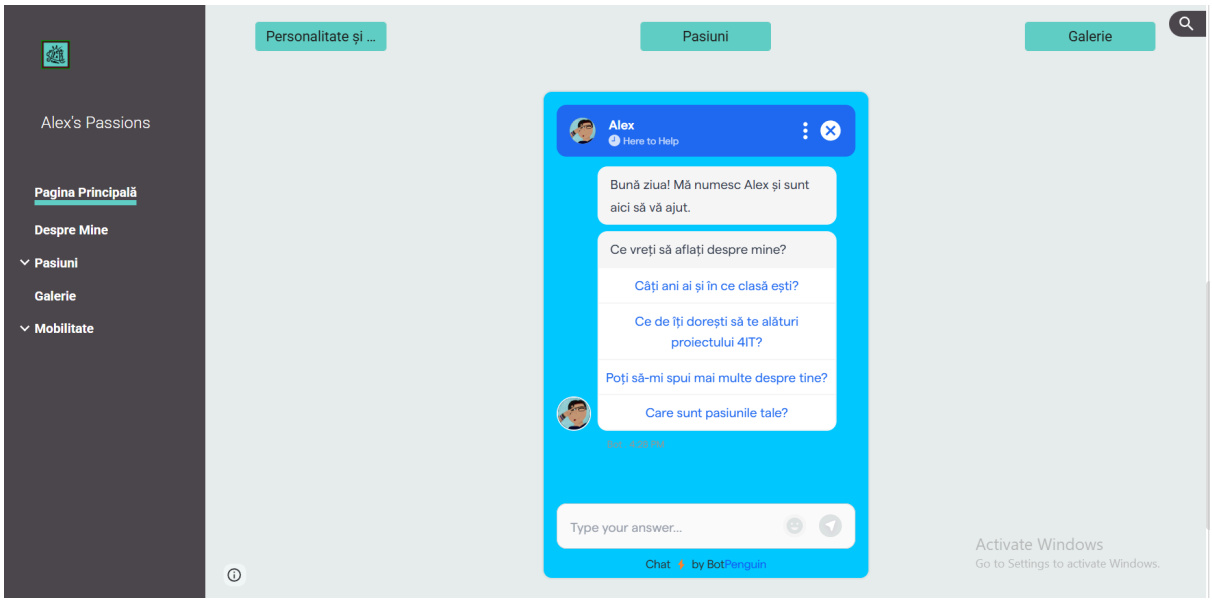
Practice tutor: Lazaros Lazarou

Participant: GHIORGHIȚĂ ALEXANDRU-CRISTIAN

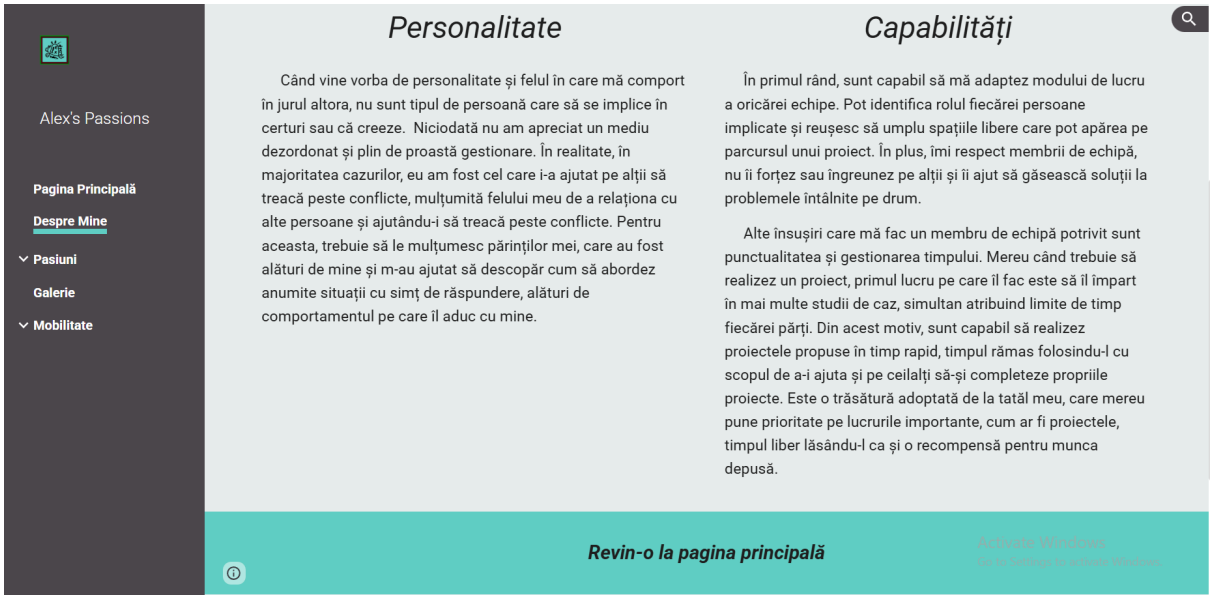
DISCLAIMER: The European Commission’s support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

ALEX’S PASSIONS






Alex's Passions
Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Cine sunt eu, cu adevărat
Personalitate
Când vine vorba de personalitate și felul în care mă comport în jurul altora, nu sunt tipul de persoană care să se implice în certuri sau că creeze. Niciodată nu am apreciat un mediu dezordonat și plin de proastă gestionare. În realitate, în majoritatea cazurilor, eu am fost cel care i-a ajutat pe alții să treacă peste conflicte, mulțumită felului meu de a relaționa cu alte persoane și ajutându-i să treacă peste conflicte. Pentru aceasta, trebuie să le mulțumesc părinților mei, care au fost alături de mine și m-au ajutat să descopăr cum să abordez anumite situații cu simț de răspundere, alături de comportamentul pe care îl aduc cu mine.
Capabilități
În primul rând, sunt capabil să mă adaptez modului de lucru a oricărei echipe. Pot identifica rolul fiecărei persoane implicate și reușesc să umplu spațiile libere care pot apărea pe parcursul unui proiect. În plus, îmi respect membrii de echipă, nu îi forțez sau îngreunez pe alții și îi ajut să găsească soluții la problemele întâlnite pe drum.
Alte însușiri care mă fac un membru de echipă potrivit sunt punctualitatea și gestionarea timpului. Mereu când trebuie să realizez un proiect, primul lucru pe care îl fac este să îl împart în mai multe studii de caz, simultan atribuind limite de timp

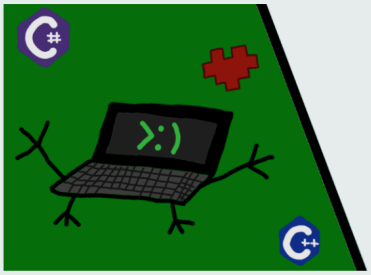

Alex's Passions
Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Personalitate
Când vine vorba de personalitate și felul în care mă comport în jurul altora, nu sunt tipul de persoană care să se implice în certuri sau că creeze. Niciodată nu am apreciat un mediu dezordonat și plin de proastă gestionare. În realitate, în majoritatea cazurilor, eu am fost cel care i-a ajutat pe alții să treacă peste conflicte, mulțumită felului meu de a relaționa cu alte persoane și ajutându-i să treacă peste conflicte. Pentru aceasta, trebuie să le mulțumesc părinților mei, care au fost alături de mine și m-au ajutat să descopăr cum să abordez anumite situații cu simț de răspundere, alături de comportamentul pe care îl aduc cu mine.
Capabilități
În primul rând, sunt capabil să mă adaptez modului de lucru a oricărei echipe. Pot identifica rolul fiecărei persoane implicate și reușesc să umplu spațiile libere care pot apărea pe parcursul unui proiect. În plus, îmi respect membrii de echipă, nu îi forțez sau îngreunez pe alții și îi ajut să găsească soluții la problemele întâlnite pe drum.
Alte însușiri care mă fac un membru de echipă potrivit sunt punctualitatea și gestionarea timpului. Mereu când trebuie să realizez un proiect, primul lucru pe care îl fac este să îl împart în mai multe studii de caz, simultan atribuind limite de timp fiecărei părți. Din acest motiv, sunt capabil să realizez proiectele propuse în timp rapid, timpul rămas folosindu-l cu scopul de a-i ajuta și pe ceilalți să-și completeze propriile proiecte. Este o trăsătură adoptată de la tatăl meu, care mereu pune prioritate pe lucrurile importante, cum ar fi proiectele, timpul liber lăsându-l ca și o recompensă pentru munca depusă.
Revin-o la pagina principală


Alex's Passions
Pagina Principală
Despre Mine
Pasiuni
Programare
Sporturi
Jurnalism
Alte Pasiuni
Galerie
Mobilitate
Pasiuni descoperite de-a lungul timpului
În ultimii ani am încercat zeci de hobby-uri cu scopul de a-mi ocupa timpul liber și de a întâlni persoane noi cu care pot conversa, simultan lucrând la starea mea fizică, cât și psihică. Multe dintre acestea au eșuat să-mi prindă atenția. În schimb, sunt câteva activități pe care le practic până și în zilele de astăzi. Dintre acestea, cele mai importante pentru mine sunt programarea în diferite limbaje, sporturile, mai exact înotul și ciclismul, și jurnalismul.
Programare
Sporturi
Jurnalism
Revin-o la pagina principală

Alex's Passions

- Pagina Principală
- Despre Mine
- Pasiuni
  - Programare
  - Sporturi
  - Jurnalism
  - Alte Pasiuni
- Galerie
- Mobilitate

Programare



Din clasa a 9-a, am început să învăț doua limbaje de programare: C++ și C#. Sunt încă la începutul călătoriei mele, dar cu timp și perseverență, o să-mi extind cunoștințele și să-i ajut pe alții să-și găsească dragostea pentru programare, indiferent de obiectivul setat.

Descoperă și celelalte pasiuni ale mele 😊

Sporturi | Jurnalism

Activate Windows
Go to Settings to activate Windows.

Alex's Passions

- Pagina Principală
- Despre Mine
- Pasiuni
  - Programare
  - Sporturi
  - Jurnalism
  - Alte Pasiuni
- Galerie
- Mobilitate

Sporturi

De când eram tânăr, am găsit o pasiune pentru sporturi. La început, mi-a luat ceva timp să aflu ce-mi place cel mai mult: fotbal, baschet, volei, tenis etc. În cele din urmă, ciclismul și înotul au devenit sporturile meu preferate, sporturi pe care le practic până în ziua de azi, când am timp și condițiile necesare.



Descoperă și celelalte pasiuni ale mele 😊

Programare | Jurnalism

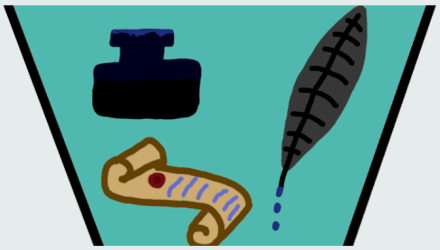
Activate Windows
Go to Settings to activate Windows.

Alex's Passions

- Pagina Principală
- Despre Mine
- Pasiuni
  - Programare
  - Sporturi
  - Jurnalism
  - Alte Pasiuni
- Galerie
- Mobilitate

Jurnalism

În vara anului 2024 am creat un jurnal. Când am timp liber, cum ar fi seara după cină, tind să scriu fie ceea ce am făcut ziua respectivă, fie gânduri pe care le-am acumulat pe o perioadă îndelungată de timp.



Pe lângă jurnalul personal, am început să lucrez și la o poveste bazată în jurul unei persoane speciale, combinată cu entuziasmul pentru Formula 1. Din septembrie, am încetat să mai lucrez la ea, dar sper ca într-o zi să fiu capabil să o duc până la sfârșit.

Descoperă și celelalte pasiuni ale mele 😊

Programare | Sporturi

Activate Windows
Go to Settings to activate Windows.

Niciodată nu sunt prea multe

Desigur, acestea nu sunt singurele pasiuni ale mele.

De exemplu, în timpul weekendului, odată la două săptămâni, am oportunitatea să vizionez unul dintre cele mai captivante și pline de emoții sporturi: Formula 1. Această pasiune a apărut la începutul lunii martie, perioadă care coincide cu începerea sezonului 2024. De-a lungul anului, am aflat despre istoria bogată a acestui motor sport, împreună cu toate personalitățile care au ieșit din acesta, cum ar fi Michael Schumacher sau Lewis Hamilton.

O altă pasiune care s-a păstrat de-a lungul anilor este cea pentru jocurile video. O majoritate din prietenii cu care vorbesc în fiecare zi i-am descoperit prin intermediul jocurilor multiplayer, dar și prin rețelele de socializare și de streaming. Persoana care mi-a făcut cunoștință cu acestea a fost sora mea mai mare, care acum lucrează în domeniul programării de jocuri video și a site-urilor profesionale.

Descoperă pasiunea mea pentru desenat

Desenatul

Activate Windows
Go to Settings to activate Windows.

Alex's Passions

Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate

Arta perseverenței

~Galerie~

Desenatul - o pasiune care a apărut simultan cu jurnalismul și la care lucrez până în prezent. Similar cu jurnalismul, acesta a părut ca metodă de exprimare a creativității, în același timp fiind o cale de a mă relaxa și de a "pierde" timpul.

Mai jos, voi prezenta o parte din lucrările mele, în ordine cronologică, alături de context și ce am învățat din fiecare.

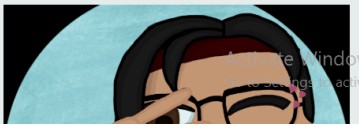


Acestea au fost primele mele desene. Scopul lor a fost de a mă acomoda cu tehnica de lucru, tableta grafică și aplicația folosită în realizarea acestora.

Primul desen reprezintă un podium din Formula 1, mai exact din Marele Premiu al Ungariei din 2024. În al doilea, este o poză cu sora mea de când am fost în Turcia în vara anului 2024, la Muzeul Panoramic din Istanbul.

Acestea au reprezentat primul pas în această pasiune a desenului.

Evident, după ce am învățat esențialele, a trebuit să trec la nivelul următor, acela fiind să-mi creez propriile desene de la zero.



Activate Windows
Go to Settings to activate Windows.

Evident, după ce am învățat esențialele, a trebuit să trec la nivelul următor, acela fiind să-mi creez propriile desene de la zero.

Dintre acestea, am decis să aleg propriul meu avatar. Deși pare simplu pentru voi, mie mi-a dovedit că sunt capabil să creez desene fără ajutor sau inspirație din altele, fapt care mi-a dat motivația să continui.

Pagina Principală





În final, dintre toate desenele pe care le-am făcut până acum, este acesta cu Max Verstappen din Formula 1(evident 😁). Este desenul cel mai detaliat dintre toate și cel la care am stat cel mai mult.

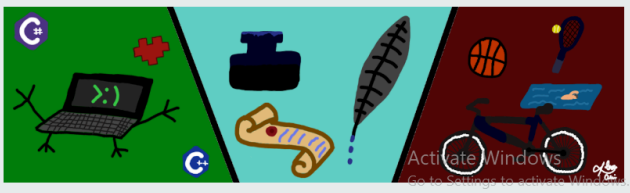
Ce am învățat din acesta a fost să am răbdare în tot ce fac, pentru ca rezultatul final să fie cât mai aproape de perfect. În plus, am descoperit cum să abordez un proiect mare, prin împărțirea acestuia în mai multe părți și completând fiecare

Activate Windows
Go to Settings to activate Windows.

Desenele create pentru acest site





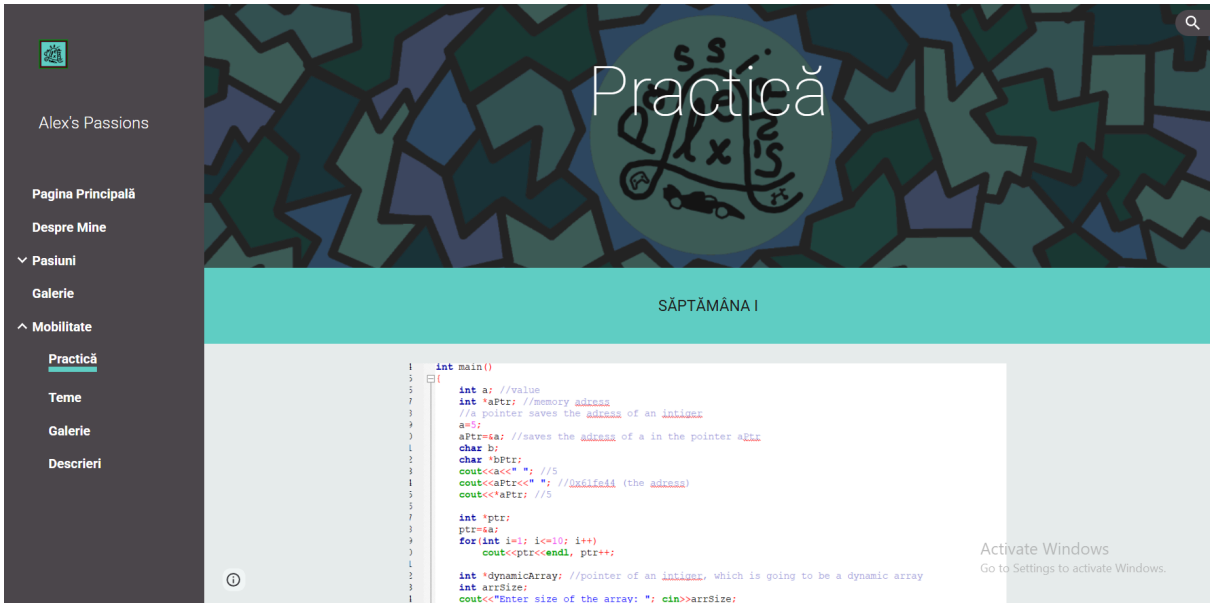


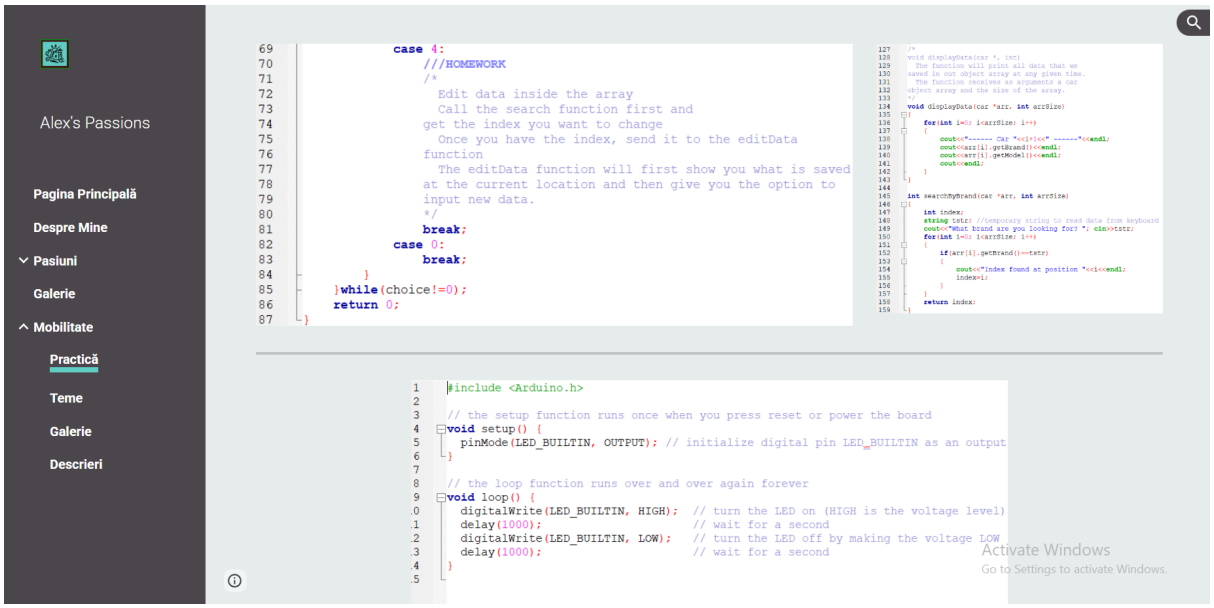
Activate Windows
Go to Settings to activate Windows.

ALEX’S PASSIONS - MOBILITATE CIPRU



ALEX'S PASSIONS - MOBILITATE CIPRU
-PRACTICA-





```
cout<<aPtr<<" "; //0x61fe44 (the adress)
cout<<*aPtr; //5

int *ptr;
ptr=&a;
for(int i=1; i<=10; i++)
    cout<<ptr<<endl, ptr++;

int *dynamicArray; //pointer of an intiger, which is going to be a dynamic array
int arrSize;
cout<<"Enter size of the array: "; cin>>arrSize;
dynamicArray=new int[arrSize]; //dynamically allocates memory for "arrSize" intigers
for(int i=1; i<=arrSize; i++) //uses the loop to fill the dynamic array with "arrSize" intigers
    dynamicArray[i]=i;
for(int i=1; i<=arrSize; i++) //displays the contents of the dynamic array
    cout<<dynamicArray[i]<<" ";
delete dynamicArray; //releases the dynamically allocated space back into the Operating System
return 0;
```

```
int main(){
    int ARRAYSIZE = 0;                          //the ARRAYSIZE is how many numbers the user wants to input. This might change if the user wishes to change it!
    int* a;                                     //a is a pointer to an integer. This is going to be our main array to save integers
    int* temp;                                  //temp is a pointer to an integer and is going to be a temporary array. We need it in order to be able to GROW our arr
    int  n = 0, tmpN=0;                         //n is the counter to count the iterations. tmpN is a number we read from the user in order to see if they need to inp

    //--- Read into the array
    cout<<"How many numbers would you like to enter?:"; //ask the user and create an array according to the user's demand.
    cin>>ARRAYSIZE;                             //read the size of the array from the user
    a = new int[ARRAYSIZE];                     //dynamically create an array with the number the user gave to us.

    do{
        cout<<"Enter number "<<n<<" ->";        //prompt the user to enter the numbers
        cin >> a[n];                            //save the number we just read into our dynamic array
        cout<<"Number saved!"<<endl;
        n++;                                    //after saving the number we have to increase the counter
        if (n >= ARRAYSIZE){                    //if the new counter is equal to the ARRAYSIZE we need to GROW THE ARRAY
            cout<<"Out of Space!!!! would you like to enter more? Please enter how many(0 to Exit): ";//LET THE USER KNOW THAT THE ARRAY IS OUT OF SPACE AND ASK HOW MANY MORE
            cin>>tmpN;                          //read how many more number the user needs to input
            if (tmpN > 0){                      //ERROR CHECKING: If the number the user gave us is not zero OR below zero we grow the array
                ARRAYSIZE = ARRAYSIZE + tmpN;   // Add to the previous size the added size the user gave us
                temp = new int[ARRAYSIZE];      // create new temporary bigger array. We need it to copy the contents of the previous array so we don't lose them
                for (int i=0; i<n; i++) {       //We use the for loop to copy the contents from the previous array into the temporary array(which has a bigger size
                    temp[i] = a[i];             // copy values to new array.
                }
                delete [] a;                    // free old array memory. ALSO SEE LAST COMMENT UNDER MAIN
                a = temp;                       // now a points to new array.
            }
        }
```

Activate Windows
Go to Settings to activate Windows.

Alex's Passions

Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Practică
Teme
Galerie
Descrieri

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

int main(int argc, char *argv[])
{
    cout<<"You have entered "<<argc<<" arguments:"<<endl;
    int i=0;
    while(i<argc)
    {
        cout<<"Argument "<<i+1<<": "<<argv[i]<<endl;
        i++;
    }
}
```

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

class student{
    private:
        string name;
        string lname;
        float exam1;
        float exam2;
    public:
        student(){//constructor: We use constructors to initialize the object's variables
            name="EMPTY";
            lname="EMPTY";
            exam1=1.1;
            exam2=1.1;
        }
    void setName(string n){  name=n; }
    void setLname(string l){ lname=l; }
    void setExam1(float e1){ exam1=e1;}
    void setExam2(float e2){ exam2=e2;}
    string getName(){ return name;}
    string getLname(){ return lname;}
    float getExam1(){ return exam1;}
    float getExam2(){ return exam2;}
};
```

```
const int MAXSTUDENTS=5;
student myclass[MAXSTUDENTS];
string tname="", tlname="";
float texam1=0.0, texam2=0.0;

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

    cout<<"Name "<<i<<": ";
    cin>>tname;
    myclass[i].setName(tname);

    cout<<"Last Name "<<i<<": ";
    cin>>tlname;
    myclass[i].setLname(tlname);
```

Alex's Passions

Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Practică
Teme
Galerie
Descrieri

```
        string lname;
        float exam1;
        float exam2;
    public:
        student(){//constructor: We use constructors to initialize the object's variables
            name="EMPTY";
            lname="EMPTY";
            exam1=1.1;
            exam2=1.1;
        }
    void setName(string n){  name=n; }
    void setLname(string l){ lname=l; }
    void setExam1(float e1){ exam1=e1;}
    void setExam2(float e2){ exam2=e2;}
    string getName(){ return name;}
    string getLname(){ return lname;}
    float getExam1(){ return exam1;}
    float getExam2(){ return exam2;}
};
```

```
        cout<<"Name "<<i<<": ";
        cin>>tname;
        myclass[i].setName(tname);

        cout<<"Last Name "<<i<<": ";
        cin>>tlname;
        myclass[i].setLname(tlname);

        cout<<"1st Exam "<<i<<": ";
        cin>>texam1;
        myclass[i].setExam1(texam1);


        cout<<"2nd Exam "<<i<<": ";
        cin>>texam2;
        myclass[i].setExam2(texam2);

    }
    cout<<endl<<endl<<endl;
    cout<<"---------YOU HAVE SAVED THE FOLLOWING INFO ----------"<<endl;
    for (int i=0;i<MAXSTUDENTS;i++){

        cout<<myclass[i].getName()<<" "<<myclass[i].getLname()<<endl;
        cout<<"Exam 1: "<<myclass[i].getExam1()<<" - Exam2: "<<myclass[i].getExam2()<<endl;
```

Alex's Passions

Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Practică
Teme
Galerie
Descrieri

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

class cars
{
    private:
        string brand;
        string model;
        string plateNum;
        string owner;
    public:
        cars()
        {
            brand="";
            model="";
            plateNum="";
            owner="";
        }
    void setBrand(string B){ brand=B; }
    void setModel(string M){ model=M; }
    void setPlateNum(string P){ plateNum=P; }
    void setOwner(string O){ owner=O; }
    string getBrand(){ return brand; }
    string getModel(){ return model; }
    string getPlateNum(){ return plateNum; }
    string getOwner(){ return owner; }
};
```

```
void add(cars carArray[], int n)
{
    string userInput,tmp2,tmp3;
    cout<<"-------- CAR "<<n<<" --------"<<endl;
    cout<<"Brand: "; cin>>userInput; carArray[n].setBrand(userInput);
    cout<<"Model: "; cin>>userInput; carArray[n].setModel(userInput);
    cout<<"Plate Number: "; cin>>userInput>>tmp2>>tmp3; //THE ">>" OPERATOR READS UP TO THE WHITE SPACE IN A SENTENCE
    userInput=userInput+" "+tmp2+" "+tmp3; carArray[n].setPlateNum(userInput); //WE USE "+' ' " TO SEPARATE THE WORDS WE READ IN THE LAST LANE
    cout<<"Owner: "; cin>>userInput; carArray[n].setOwner(userInput);
}

///add data, show data, search by model, price, owner, etc. After, convert to growing dynamic arrays
```

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

///pass a pointer to a function that will point to an array and do a search

int function(int *x, int v[], int n)
{
    int ok=0;
    for(int i=1; i<=n&&!ok; i++)
        if(*x==v[i])
        {
            return 1;
        }
    return 0;
}

int main()
{
    int v[101],n,x;
    cin>>n;
    for(int i=1; i<=n; i++)
        cin>>v[i];
    int *xPtr;
    cout<<"Number you want to search for: "; cin>>x;
    xPtr=&x;

    int *xPtr;
    cout<<"Number you want to search for: "; cin>>x;
    xPtr=&x;
    if(function(xPtr,v,n)) cout<<"Number "<<x<<" was found.";
    else cout<<"Number "<<x<<" was NOT found.";
    return 0;
}
```

Activate Windows
Go to Settings to activate Windows.

Alex's Passions

Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Practică
Teme
Galerie
Descrieri

SĂPTĂMÂNA II

```
int main()
{
    int MAXCARS=0;
    cars *carArray=new cars[MAXCARS+1];
    int choice,ok=0;
    while(!ok)
    {
        int p=0;
        menu(); cin>>choice;
        if(choice==1)
        {
            MAXCARS++;
            cars *tempN=new cars[MAXCARS+1];
            for(int i=1; i<=MAXCARS-1; i++)
                tempN[i]=carArray[i];
            add(tempN,MAXCARS);
            delete []carArray;
            carArray=tempN;
            p=1;
        }
        else if(choice==2) show(carArray,MAXCARS),p=1;
        else if(choice==3) c3(carArray,MAXCARS),p=1;
        else if(choice==4) c4(carArray,MAXCARS),p=1;
        else if(choice==5) c5(carArray,MAXCARS),p=1;
        else if(choice==6) c6(carArray,MAXCARS),p=1;
        else if(choice==0) ok=1,p=1;
        if(p==0) cout<<"Car not found based on information given!";
        cout<<endl<<endl;
    }
    delete carArray;
    return 0;
}

///HOMEWORK: SUPERMARKET EXAMPLE
```

```
class cars
{
    private:
        string brand;
        string model;
        string plateNum;
        string owner;
    public:
        cars()
        {
            brand="";
            model="";
            plateNum="";
            owner="";
        }
    void setBrand(string B){ brand=B; }
    void setModel(string M){ model=M; }
    void setPlateNum(string P){ plateNum=P; }
    void setOwner(string O){ owner=O; }
    string getBrand(){ return brand; }
    string getModel(){ return model; }
    string getPlateNum(){ return plateNum; }
    string getOwner(){ return owner; }
};

void display(cars carArray[], int i)
{
    cout<<endl;
    cout<<"------- CAR "<<i<<" -------"<<endl;
    cout<<"Brand: "; cout<<carArray[i].getBrand()<<endl;
    cout<<"Model: "; cout<<carArray[i].getModel()<<endl;
    cout<<"Plate Number: "; cout<<carArray[i].getPlateNum()<<endl;
    cout<<"Owner: "; cout<<carArray[i].getOwner()<<endl;
    cout<<endl;
}
```

Activate Windows
Go to Settings to activate Windows.

Alex's Passions

Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Practică
Teme
Galerie
Descrieri

```
void menu()
{
    cout<<"1. Add information"<<endl;
    cout<<"2. Display all"<<endl;
    cout<<"3. Search by brand"<<endl;
    cout<<"4. Search by model"<<endl;
    cout<<"5. Search by plate number"<<endl;
    cout<<"6. Search by owner"<<endl;
    cout<<"0. Exit"<<endl;
    cout<<"Your Choice: ";
}

void add(cars carArray[], int n)
{
    string userInput="",tmp2="",tmp3="";
    cout<<"------- CAR "<<n<<" -------"<<endl;
    cout<<"Brand: "; cin>>userInput; carArray[n].setBrand(userInput);
    cout<<"Model: "; cin>>userInput; carArray[n].setModel(userInput);
    cout<<"Plate Number: "; cin>>userInput>>tmp2>>tmp3; //USE ">>" OPERATOR READS UP TO THE WHITE SPACE IN A SENTENCE
    userInput=userInput+" "+tmp2+" "+tmp3; carArray[n].setPlateNum(userInput); //WE USE "+' ' " TO SEPARATE THE WORDS WE READ IN THE LAST LANE
    cout<<"Owner: "; cin>>userInput; carArray[n].setOwner(userInput);
}

void show(cars carArray[], int MAXCARS)
{
    for(int i=1; i<=MAXCARS; i++)
        display(carArray,i);
}

void c3(cars carArray[], int MAXCARS)
{
    string userInput;
    cout<<"Brand you are searching for: "; cin>>userInput;
    for(int i=1; i<=MAXCARS; i++)
        if(userInput==carArray[i].getBrand()) display(carArray,i);
}
```

```
void c3(cars carArray[], int MAXCARS)
{
    string userInput;
    cout<<"Brand you are searching for: "; cin>>userInput;
    for(int i=1; i<=MAXCARS; i++)
        if(userInput==carArray[i].getBrand()) display(carArray,i);
}

void c4(cars carArray[], int MAXCARS)
{
    string userInput;
    cout<<"Model you are searching for: "; cin>>userInput;
    for(int i=1; i<=MAXCARS; i++)
        if(userInput==carArray[i].getModel()) display(carArray,i);
}

void c5(cars carArray[], int MAXCARS)
{
    string userInput,tmp2,tmp3;
    cout<<"Plate you are searching for: "; cin>>userInput>>tmp2>>tmp3;
    userInput=userInput+" "+tmp2+" "+tmp3;
    for(int i=1; i<=MAXCARS; i++)
        if(userInput==carArray[i].getPlateNum()) display(carArray,i);
}

void c6(cars carArray[], int MAXCARS)
{
    string userInput;
    cout<<"Owner you are searching for: "; cin>>userInput;
    for(int i=1; i<=MAXCARS; i++)
        if(userInput==carArray[i].getOwner()) display(carArray,i);
}
```

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

//CREATE THE CLASS
class car{
    private:
        string brand;
        string model;
    public:

        car(){
            brand="EMPTY";
            model="EMPTY";

        }
        void setBrand(string B){
            brand=B;
        }
        void setModel(string M){
            model=M;
        }
        string getBrand(){
```

```
int main(){

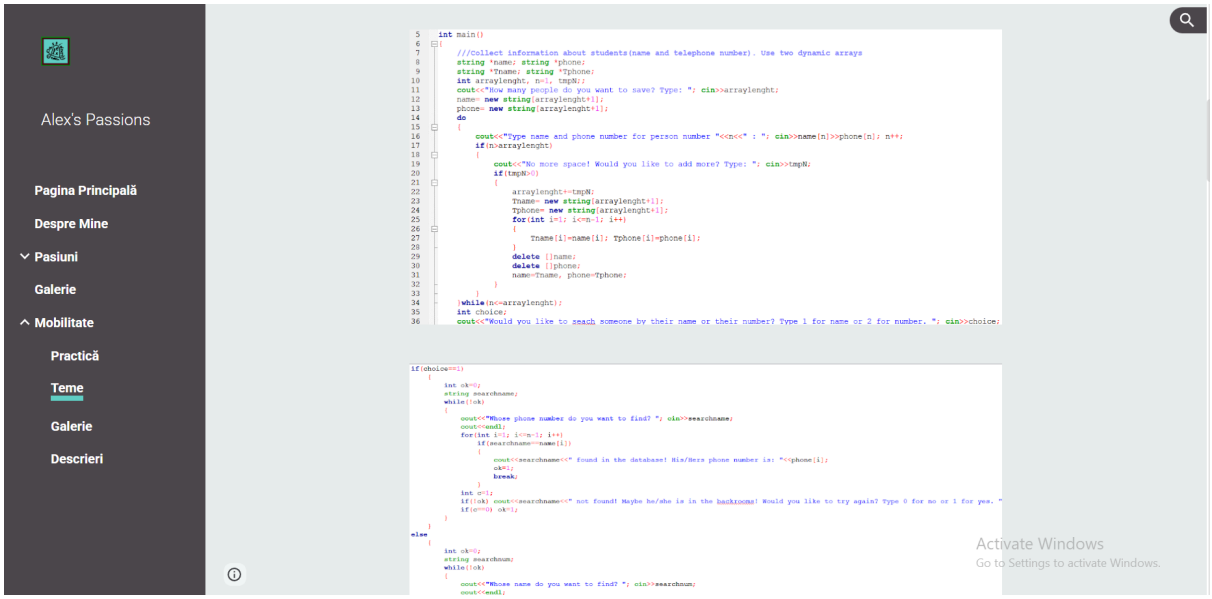
    int ARRAYSIZE=4;
    int nextAvailPos=0;
    car *carsArray;
    carsArray=new car[ARRAYSIZE];
    int choice=-1;
    int searchIndex=-1;
    do
    {
        cout<<"Please Choose from the Menu Below: "<<endl;
        cout<<"1. Enter Data"<<endl;
        cout<<"2. Display Data"<<endl;
        cout<<"3. Search by Brand"<<endl;
        cout<<"0. Exit"<<endl;
        cin>>choice;
        switch(choice)
        {
            case 1:
                carsArray=readData(carsArray,ARRAYSIZE,nextAvailPos);
                break;
            case 2:
                /*
                  Because we are growing the array dynamically the
                  variable ARRAYSIZE is the TOTAL SIZE OF THE ARRAY,
                  which may contain EMPTY positions.
                  The variable nextAvailPos tells us the last index
                  which contains data
                */
                displayData(carsArray,nextAvailPos);
                break;
```

```
/*
function car *readData(car *, int)
  This function populates the array with information
from the user
  The function accepts a pointer to a car object(the array)
and the size of the array
  The function returns back a pointer to a car object which is
an array and is now populated with data
*/
car *readData(car *arr, int arrSize, int nextPos)
{
  int numCars=0;
  int newSize; //The new size of the array which is calculated by adding arraysize+numCars
  string tString; //temporary string to read keyboard input
  car *tmp; //temporary car object to copy everything from the car array and grow the array
  cout<<"How many cars are you entering? ";
  cin>>numCars; //read from the keyboard how many cars does the user want to input
  for(int i=0; i<numCars; i++) //repeat the reading process "numCars" size
  {
    if(nextPos>=arrSize) //first check if the array is full
    {
      newSize=arrSize+numCars; //calculates the new size
      tmp=new car[newSize]; //allocates temporary space for the new bigger array
      for(int j=0; j<arrSize; j++)
        tmp[j]=arr[j]; //copies everything from the old array to the bigger one
      arrSize=newSize; //saves the new size to arraysize (update the array size)
      delete []arr; //release the memory for the smaller array
      arr=tmp; //set the pointer of car array to point to the bigger array
    }
    cout<<"Enter BRAND position for car "<<nextPos+1<<": ";
    cin>>tString;
    arr[nextPos].setBrand(tString);
```

Activate Windows
Go to Settings to activate Windows.

ALEX'S PASSIONS - MOBILITATE CIPRU
-TEMA-





```
        if(!ok) cout<<searchname<<" not found! Maybe he/she is in the backrooms! Would you like to try again? Type 0 for no or 1 for yes. ";
        if(c==0) ok=1;
    }
}
else
    {
        int ok=0;
        string searchnum;
        while(!ok)
        {
            cout<<"Whose name do you want to find? "; cin>>searchnum;
            cout<<endl;
            for(int i=1; i<=n-1; i++)
                if(searchnum==phone[i])
                {
                    cout<<searchnum<<" found in the database! His/Hers name is: "<<name[i];
                    ok=1;
                    break;
                }
            int c=1;
            if(!ok) cout<<searchnum<<" not found! Maybe he/she is in the backrooms! Would you like to try again? Type 0 for no or 1 for yes. ";
            if(c==0) ok=1;
        }
    }
```

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

class cars
{
    private:
        string brand;
        string model;
        string plate;
        string owner;
    public:
        void setBrand(string b){ brand=b; }
        void setModel(string m){ model=m; }
        void setPlate(string p){ plate=p; }
        void setOwner(string o){ owner=o; }
        string getBrand(){ return brand; }
        string getModel(){ return model; }
        string getPlate(){ return plate; }
        string getOwner(){ return owner; }
```

```
int main(){
    int MAXCARS;
    cout<<"How many cars do you want to save? "; cin>>MAXCARS;
    cars myclass[MAXCARS+1];
    string tbrand="",tmodel="",tplate="",towner="";
    for (int i=1;i<=MAXCARS;i++)
        {
            cout<<"Brand "<<i<<": ";
            cin>>tbrand;
            myclass[i].setBrand(tbrand);
            cout<<"Model "<<i<<": ";
            cin>>tmodel;
            myclass[i].setModel(tmodel);
            cout<<"Car Plate "<<i<<": ";
            cin>>tplate;
            myclass[i].setPlate(tplate);
            cout<<"Owner "<<i<<": ";
            cin>>towner;
            myclass[i].setOwner(towner);
            cout<<endl<<endl;
        }
    cout<<endl<<endl<<endl;
    for (int i=1;i<=MAXCARS;i++)
    {
        cout<<"INFORMATION FOR CAR NUMBER "<<i<<":"<<endl;
        cout<<"Car: "<<myclass[i].getBrand()<<" "<<myclass[i].getModel()<<endl;
        cout<<"Car Plate: "<<myclass[i].getPlate()<<"   Car Owner: "<<myclass[i].getOwner();
        cout<<endl<<endl<<endl;
    }
    return 0;
}
```

Activate Windows
Go to Settings to activate Windows.


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

class product
{
    private:
        string brand;
        string type;
        int price;
        int stock;
        long int barcode;
    public:
        void setBrand(string b){ brand=b; }
        void setType(string t){ type=t; }
        void setPrice(int p){ price=p; }
        void setStock(int s){ stock=s; }
        void setBarcode(long int bc){ barcode=bc; }
        string getBrand(){ return brand; }
        string getType(){ return type; }
        int getPrice(){ return price; }
        int getStock(){ return stock; }
        long int getBarcode(){ return barcode; }
};

string tbrand,ttype;
int tprice,tstock;
long int tbarcode;
```

```
void menu()
{
    cout<<"1. Add information."<<endl;
    cout<<"2. Display all."<<endl;
    cout<<"3. Search by Brand."<<endl;
    cout<<"4. Search by Type."<<endl;
    cout<<"5. Search by Price."<<endl;
    cout<<"6. Search by Stock."<<endl;
    cout<<"0. Exit"<<endl;
    cout<<"Your option: ";
}

void add(product myclass[], int i)
{
    cout<<"Brand and type of product number "<<i<<": ";
    cin>>tbrand>>ttype;
    myclass[i].setBrand(tbrand); myclass[i].setType(ttype);
    cout<<"Set price and stock remaining: ";
    cin>>tprice>>tstock;
    myclass[i].setPrice(tprice); myclass[i].setStock(tstock);
    cout<<"Set barcode for product number: "<<i<<" : ";
    cin>>tbarcode;
    myclass[i].setBarcode(tbarcode);
    cout<<endl<<endl;
}
```

```
void display(product myclass[], int n)
{
    cout<<"Brand "<<n<<": "<<myclass[n].getBrand()<<endl;
    cout<<"Type "<<n<<": "<<myclass[n].getType()<<endl;
    cout<<"Price "<<n<<": "<<myclass[n].getPrice()<<endl;
    cout<<"Stock "<<n<<": "<<myclass[n].getStock()<<endl;
    cout<<"Barcode "<<n<<": "<<myclass[n].getBarcode()<<endl;
```

```
void c5(product myclass[], int MAXPROD)
{
    cout<<"Price you are searching for: "; cin>>tprice;
    for(int i=1; i<=MAXPROD; i++)
        if(tprice==myclass[i].getPrice())
            cout<<myclass[i].getBarcode()<<"    ";
```



```
    cout<<"Brand "<<n<<": "<<myclass[n].getBrand()<<endl;
    cout<<"Type "<<n<<": "<<myclass[n].getType()<<endl;
    cout<<"Price "<<n<<": "<<myclass[n].getPrice()<<endl;
    cout<<"Stock "<<n<<": "<<myclass[n].getStock()<<endl;
    cout<<"Barcode "<<n<<": "<<myclass[n].getBarcode()<<endl;
    cout<<endl<<endl;
}

void c3(product myclass[], int MAXPROD)
{
    cout<<"Brand you are searching for: "; cin>>tbrand;
    for(int i=1; i<=MAXPROD; i++)
        if(tbrand==myclass[i].getBrand())
            cout<<myclass[i].getBarcode()<<"    ";
    cout<<endl<<endl;
}

void c4(product myclass[], int MAXPROD)
{
    cout<<"Type you are searching for: "; cin>>ttype;
    for(int i=1; i<=MAXPROD; i++)
        if(ttype==myclass[i].getType())
            cout<<myclass[i].getBarcode()<<"    ";
    cout<<endl<<endl;
}
```

```
    cout<<"Price you are searching for: "; cin>>tprice;
    for(int i=1; i<=MAXPROD; i++)
        if(tprice==myclass[i].getPrice())
            cout<<myclass[i].getBarcode()<<"    ";
    cout<<endl<<endl;
}

void c6(product myclass[], int MAXPROD)
{
    cout<<"Stock you are searching for: "; cin>>tstock;
    for(int i=1; i<=MAXPROD; i++)
        if(tstock==myclass[i].getStock())
            cout<<myclass[i].getBarcode()<<"    ";
    cout<<endl<<endl;
}
```

```
int MAXPROD=0;
product *myclass=new product[MAXPROD+1];
int choice; int ok=0;
do
{
    menu(); cin>>choice;
    if(choice==1)
    {
        MAXPROD++;
        product *tempN=new product[MAXPROD+1];
        for(int i=1; i<=MAXPROD-1; i++)
            tempN[i]=myclass[i];
        add(tempN,MAXPROD);
        delete []myclass;
        myclass=tempN;
    }
    else if(choice==2)
```

Activate Windows
Go to Settings to activate Windows.

```
        product *tempN=new product[MAXPROD+1];
        for(int i=1; i<=MAXPROD-1; i++)
            tempN[i]=myclass[i];
        add(tempN,MAXPROD);
        delete []myclass;
        myclass=tempN;
    }
    else if(choice==2)
    {
        for(int i=1; i<=MAXPROD; i++)
            display(myclass,i);
    }
    else if(choice==3) c3(myclass,MAXPROD);
    else if(choice==4) c4(myclass,MAXPROD);
    else if(choice==5) c5(myclass,MAXPROD);
    else if(choice==6) c6(myclass,MAXPROD);
    else if(choice==0) ok=1;
}while(!ok);
return 0;
}
```

```
//CREATE THE CLASS
class car{
    private:
        string brand;
        string model;
    public:

        car(){
            brand="EMPTY";
            model="EMPTY";

        }
        void setBrand(string B){
            brand=B;
        }
        void setModel(string M){
            model=M;
        }
        string getBrand(){
            return brand;

            return brand;
        }
        string getModel(){
            return model;
        }
};

car *readData(car *, int &, int &);
void displayData(car *, int);
int searchByBrand(car *, int);
void editData(car *, int);
```

```
int main(){

    int ARRAYSIZE=4;
    int nextAvailPos=0;
    car *carsArray;
    carsArray=new car[ARRAYSIZE];
    int choice=-1;
    int searchIndex=-1;
    do
    {
        cout<<"Please Choose from the Menu Below: "<<endl;
        cout<<"1. Enter Data"<<endl;
        cout<<"2. Display Data"<<endl;
        cout<<"3. Search by Brand"<<endl;
        cout<<"4. Modify Data based on Brand"<<endl;
        cout<<"0. Exit"<<endl;
        cin>>choice;
        switch(choice)

        switch(choice)
        {
            case 1:
                carsArray=readData(carsArray,ARRAYSIZE,nextAvailPos);
                break;
```

Activate Windows
Go to Settings to activate Windows.

Alex's Passions

Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Practică
Teme
Galerie
Descrieri

```
        switch(choice)
        {
            case 1:
                carsArray=readData(carsArray,ARRAYSIZE,nextAvailPos);
                break;
            case 2:
                /*
                  Because we are growing the array dynamically the
                variable ARRAYSIZE is the TOTAL SIZE OF THE ARRAY,
                which may contain empty positions.
                  The variable nextAvailPos tells us the last index
                which contains data
                */
                displayData(carsArray,nextAvailPos);
                break;
            case 3:
                searchIndex=searchByBrand(carsArray,nextAvailPos);
                break;
            case 4:
                editData(carsArray,nextAvailPos);
                break;
            case 0:
                break;
        }
    }while(choice!=0);
    return 0;
}
```

```
car *readData(car *arr, int &arrSize, int &nextPos)
{
    int numCars=0;
    int newSize; //the new size of the array which is calculated by adding arraySize+numCars
    string tString; //temporary string to read keyboard input
    car *tmp; //temporary car object to copy everything from the car array and grow the array
    cout<<"How many cars are you entering? ";
    cin>>numCars; //read from the keyboard how many cars does the user want to input
    for(int i=0; i<numCars; i++) //repeat the reading process "numCars" size
    {
        if(nextPos>=arrSize) //first check if the array is full
        {
            newSize=arrSize+numCars; //calculates the new size
            tmp=new car[newSize]; //allocates temporary space for the new bigger array
            for(int j=0; j<arrSize; j++)
                tmp[j]=arr[j]; //copies everything from the smaller array to the bigger one
            arrSize=newSize; //saves the new size to arraySize (update the array size)
            delete []arr; //release the memory from the old array back to the OS
            arr=tmp; //set the pointer of the array to point to the new allocated space
        }
        cout<<"Enter BRAND position for car "<<nextPos+1<<": ";
        cin>>tString;
        arr[nextPos].setBrand(tString);
        cout<<"Enter MODEL position for car "<<nextPos+1<<": ";
        cin>>tString;
        arr[nextPos].setModel(tString);
        nextPos++;
    }
    return arr;
```

```
saved in out object array at any given time.
  The function receives as arguments a car
object array and the size of the array.
*/
void displayData(car *arr, int arrSize)
```

```
void editData(car *arr, int arrSize)
{
    int ch=-1;
    string tstr;
```

Alex's Passions

Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Practică
Teme
Galerie
Descrieri

```
                displayData(carsArray,nextAvailPos);
                break;
            case 3:
                searchIndex=searchByBrand(carsArray,nextAvailPos);
                break;
            case 4:
                editData(carsArray,nextAvailPos);
                break;
            case 0:
                break;
        }
    }while(choice!=0);
    return 0;
}
```

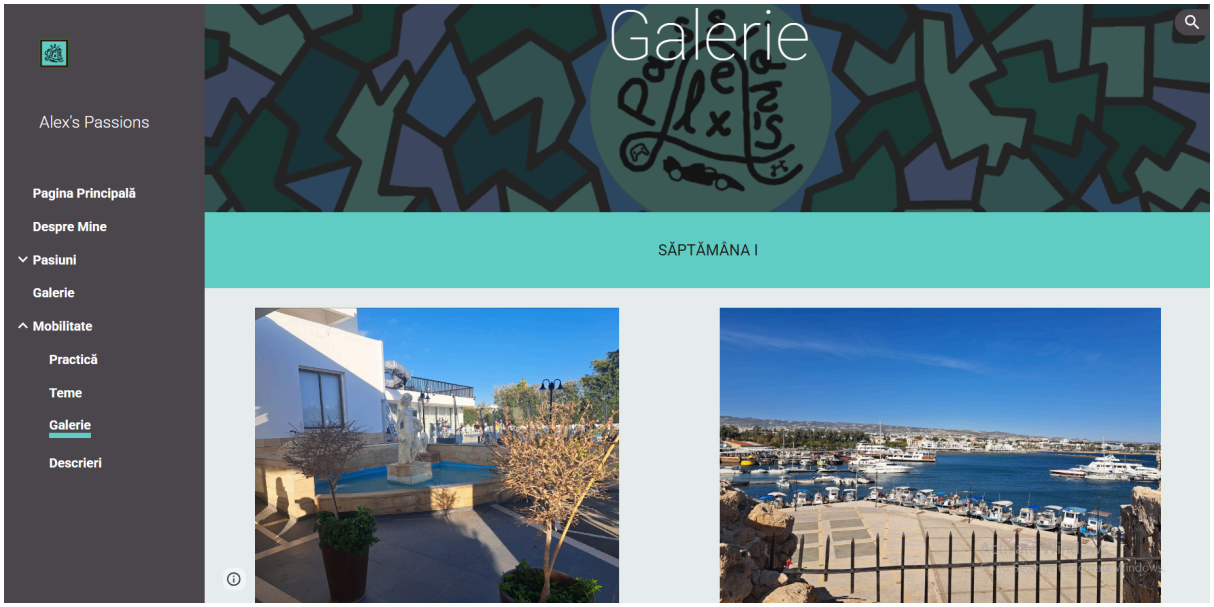
```
            delete []arr; //release the memory from the old array back to the OS
            arr=tmp; //set the pointer of the array to point to the new allocated space
        }
        cout<<"Enter BRAND position for car "<<nextPos+1<<": ";
        cin>>tString;
        arr[nextPos].setBrand(tString);
        cout<<"Enter MODEL position for car "<<nextPos+1<<": ";
        cin>>tString;
        arr[nextPos].setModel(tString);
        nextPos++;
    }
    return arr;
```

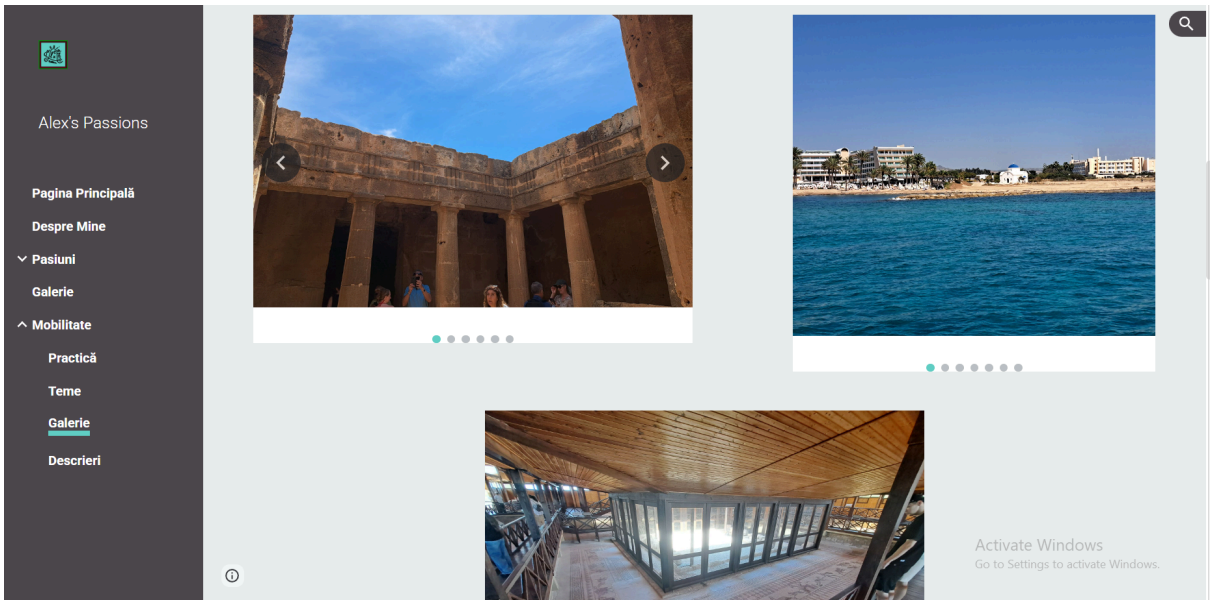
```
saved in out object array at any given time.
  The function receives as arguments a car
object array and the size of the array.
*/
void displayData(car *arr, int arrSize)
{
    for(int i=0; i<arrSize; i++)
    {
        cout<<"------ Car "<<i+1<<" ------"<<endl;
        cout<<arr[i].getBrand()<<endl;
        cout<<arr[i].getModel()<<endl;
        cout<<endl;
    }
}

int searchByBrand(car *arr, int arrSize)
{
    int index;
    string tstr; //temporary string to read data from keyboard
    cout<<"What brand are you looking for? "; cin>>tstr;
    for(int i=0; i<arrSize; i++)
    {
        if(arr[i].getBrand()==tstr)
        {
            cout<<"Index found at position "<<i+1<<endl;
            index=i;
        }
    }
    return index;
```

```
void editData(car *arr, int arrSize)
{
    int ch=-1;
    string tstr;
    int index=searchByBrand(arr,arrSize);
    cout<<"INFOMATION ABOUT CAR "<<index+1<<endl;
    cout<<"Brand: "<<arr[index].getBrand()<<endl;
    cout<<"Model: "<<arr[index].getModel()<<endl;
    cout<<"What do you want to change? Type 1 for Brand, 2 for Model."; cin>>ch;
    cout<<endl;
    if(ch==1)
    {
        cout<<"What will be the new Brand? "; cin>>tstr;
        arr[index].setBrand(tstr);
    }
    else if(ch==2)
    {
        cout<<"What will be the new Model? "; cin>>tstr;
        arr[index].setModel(tstr);
    }
    cout<<endl<<"Information successfully modified!"<<endl;
}
```

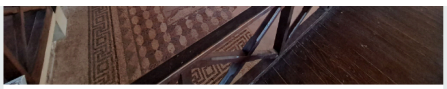
ALEX’S PASSIONS - MOBILITATE CIPRU
-GALERIE-





Alex's Passions

Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Practică
Teme
Galerie
Descrieri



SĂPTĂMÂNA II





Activate Windows
Go to Settings to activate Windows.

Alex's Passions

Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Practică
Teme
Galerie
Descrieri





Activate Windows
Go to Settings to activate Windows.

Alex's Passions

Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Practică
Teme
Galerie
Descrieri



CERTIFICATE OF RECOGNITION

ORGANIZATION FOR PROMOTION OF EUROPEAN ISSUES
hereby certifies that

Ghiorghita Alexandru-Cristian

from
Colegiul National „Roman Voda"
Bacau, Romania
successfully participated in the

4IT ERASMUS+ PROJECT
2024-1-RO01-KA121-VET-000214264

mobility period: 13.04.2025 to 27.04.2025,
in Paphos, Cyprus

ACTIVITIES CARRIED OUT:
- 60 hours of IT practical training in C++ & Arduino (14.04.2024-25.04.2025) at Lazarou ICT Professionals, tutor Lazaros Lazarou;
- 10-hour Greek Course: level pre-A1, teacher Stavroula Metaxa;
- Social & Cultural Programme, mentor Demetra Pinaretou.

Michalis Papatheraponlos,
President of
ORGANIZATION FOR PROMOTION OF EUROPEAN ISSUES

Paphos, Cyprus

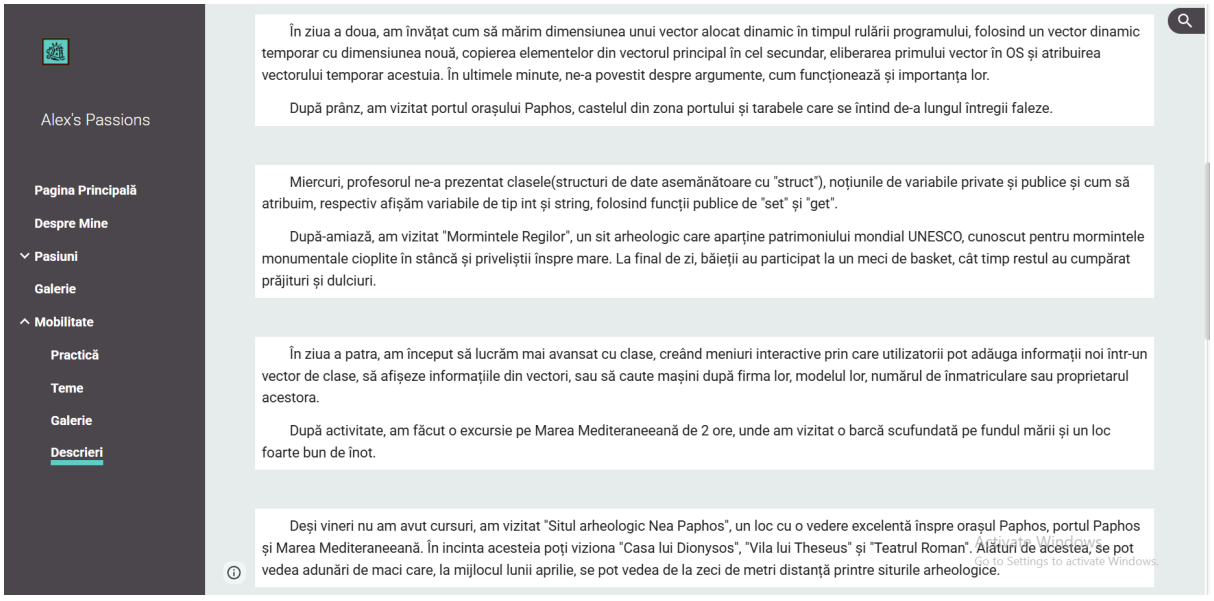
27.04.2025



Activate Windows
Go to Settings to activate Windows.

ALEX’S PASSIONS - MOBILITATE CIPRU
-DESCRIERE-





SĂPTĂMÂNA II

Pe 21 aprilie 2025, am creat programe în care am folosit meniuri de navigare, funcții pentru realizarea căutărilor și mărirea vectorilor alocați dinamic, folosind funcții care returnează un pointer. Ca temă, am avut de modificat o temă anterioară încât să folosim ce am învățat azi la practică.

Pe 22 aprilie, am continuat să lucrăm cu meniuri interactive, prin care putem adăuga informații folosind vectori alocați dinamic, afișarea informațiilor și căutarea după mai multe criterii.

După-amiază, am vizitat centrul vechi al orașului Paphos, un loc plin de istorie și turiști, cât și de restaurante în stil tradițional și magazine cu suveniruri cipriote și turcești.

Cursul de miercuri, 23 aprilie, a fost împărțit în două jumătăți. În prima parte, am continuat programul pe care l-am început ieri și am adăugat la el o funcție care să caute indexul unui element pe baza unui criteriu. În a doua jumătate, mentorul ne-a prezentat pe scurt Arduino, ce este, și noțiunile de baza pe care le vom învăța mâine.

După cursuri, am vizitat centrul vechi, de data asta cu ghid, care ne-a prezentat pe scurt istoria acelui loc, modul în care coloniile otomane și britanice au afectat piața și procesul de transformare într-un spațiu turistic de vizitat.

Activate Windows
Go to Settings to activate Windows.

În penultima zi, mentorul ne-a prezentat componentele unui cod Arduino, funcția "setup" și funcția "loop", noțiunile de LOW și HIGH, care

Alex's Passions

Pagina Principală
Despre Mine
Pasiuni
Galerie
Mobilitate
Practică
Teme
Galerie
Descrieri

După-amiază, am vizitat centrul vechi al orașului Paphos, un loc plin de istorie și turiști, cât și de restaurante în stil tradițional și magazine cu suveniruri cipriote și turcești.

Cursul de miercuri, 23 aprilie, a fost împărțit în două jumătăți. În prima parte, am continuat programul pe care l-am început ieri și am adăugat la el o funcție care să caute indexul unui element pe baza unui criteriu. În a doua jumătate, mentorul ne-a prezentat pe scurt Arduino, ce este, și noțiunile de baza pe care le vom învăța mâine.

După cursuri, am vizitat centrul vechi, de data asta cu ghid, care ne-a prezentat pe scurt istoria acelui loc, modul în care coloniile otomane și britanice au afectat piața și procesul de transformare într-un spațiu turistic de vizitat.

În penultima zi, mentorul ne-a prezentat componentele unui cod Arduino, funcția "setup" și funcția "loop", noțiunile de LOW și HIGH, care înseamnă închis sau 0, respectiv deschis sau 1, și un cod simplu de aprindere și închidere al unui LED. În plus, ne-a prezentat anumite tipuri de motoare și practicalitatea acestora. După aceea, am învățat ce este un circuit board, cum funcționează și din ce este alcătuit, dar și cum să creezi unul acasă.

25 aprilie 2025, ultima zi de cursuri. Această zi a fost una de emoții și de rămas bun. Ne-au fost acordate certificatele, am dat mâna cu domnul Lazaros Lazarou și am făcut poze de grup. Mentorul ne-a urat succes pe viitor și drum bun înapoi spre casă.

În final, am fost toți la grădina zoologică, unde am văzut sute de specii de păsări, mai mari și mai mici, cât și animale din toate colțurile lumii: maimuțe, șerpi, feline, târâtoare, orice. Deși aveam oportunitatea să ne jucăm cu lemurii, nu am avut timp suficient.

Activate Windows
Go to Settings to activate Windows.