

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: Dobos Erika

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.

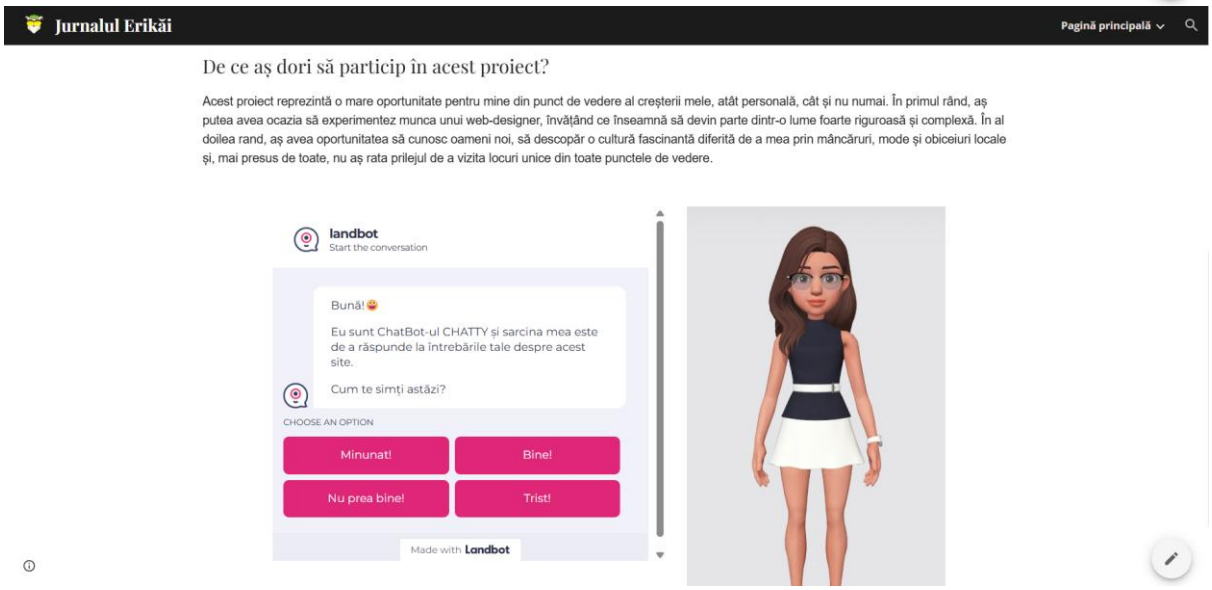
Dobos Erika

Jurnalul Erikăi - Website



Bună ziua tuturor!

Mă numesc Doboș Erika, am 16 ani și sunt o elevă a Colegiului National "Roman-Vodă".
Consider că mă pot defini ca o persoană hotărâtă, cu o dorință pasională de a învăța lucruri noi în fiecare zi pentru a face din viitorul meu un loc fără diferențe de niciun fel și pentru a putea contribui la o viață liniștită și lipsită de griji pentru cei pe care îi voi întâlni cu trecerea timpului.



De ce aș dori să particip în acest proiect?

Acest proiect reprezintă o mare oportunitate pentru mine din punct de vedere al creșterii mele, atât personală, cât și nu numai. În primul rând, aș putea avea ocazia să experimentez munca unui web-designer, învățând ce înseamnă să devin parte dintr-o lume foarte riguroasă și complexă. În al doilea rand, aș avea oportunitatea să cunosc oameni noi, să descopăr o cultură fascinantă diferită de a mea prin mâncăruri, mode și obiceiuri locale și, mai presus de toate, nu aș rata prilejul de a vizita locuri unice din toate punctele de vedere.

Dobos Erika

Jurnalul Erikăi

Pagină principală

Personalitate



De când eram mică am fost timidă și rezervată, acestea fiind o parte a caracterului meu care nu îmi plac foarte mult, pentru că nu mi-au permis să mă încred prea mult în oamenii din jurul meu. Din alt punct de vedere, însă, cred că există și câteva lucruri pozitive, și anume faptul că aceste caracteristici ale mele mi-au permis întotdeauna să fiu o persoană de încredere, asigurându-mă că cei din jurul meu se pot simți în siguranță să-mi încredințeze preocupările și gândurile lor.
Trebuie să recunosc totuși că în ultimii ani, am început să mă deschid puțin mai mult și, deși grupul meu de prieteni este destul de mic, nu am nicio problemă să mă relaționez cu alți oameni și să cunosc personalități noi și unice.

Jurnalul Erikăi

Pagină principală

În ceea ce privește caracterul meu, pot spune și că sunt o persoană sensibilă și empatică. Încerc mereu să mă pun în pielea celor pe care îi cunosc, mai ales în cele mai grele momente, și încerc să ajut cât pot de bine.
În mediile școlare și extracurriculare, sunt o persoană competitivă și încerc mereu să mă perfecționez, învățând tot ce este posibil de la cei care sunt mai buni decât mine și depășindu-mi limitele de fiecare dată când am ocazia, încercând să demonstrez că pot fi un persoană pregătită și cu o dorință infinită de a descoperi noi informații în fiecare zi.
Sunt o persoană prudentă și atentă, acționând întotdeauna cu precauție și evaluând consecințele acțiunilor mele, pentru a evita orice riscuri inutile. În ciuda faptului că sunt demnă de încredere și grijulie, trebuie să recunosc că sunt extrem de tenace, făcând deseori totul în felul meu și neascultând sfaturile din jurul meu.
În același timp, îmi umplu mereu capul cu gânduri și emoții care de multe ori sunt inutile, dar vin doar pentru că sunt foarte agitată și nu mă pot controla.



Sinceră să fiu, sunt o fată irascibilă, pentru că uneori nu accept judecăți negative despre mine, ceea ce mă face și susceptibilă. Trebuie să recunosc că este o parte a personalității mele pe care încerc să o eliberez cât mai puțin, dar uneori nu mă pot abține și îmi arăt partea cea mai slabă. Acest lucru mă face să mă simt rău, nu accept faptul că am slăbiciuni, dar muncesc din greu din acest punct de vedere și înțeleg că uneori ai nevoie de momente proaste pentru a cunoaște noi laturi ale caracterului și să înțeleg că toate acestea sunt umane.

Jurnalul Erikăi

Pagină principală

Pasiuni





Dobos Erika

Jurnalul Erikăi

Pagină principală

CĂLĂTORIA

A călători pentru mine este sinonim cu libertate, creștere, experiență, schimbare și curiozitate. Această pasiune îmi permite să cresc din multe puncte de vedere: atât personal, cât și cultural, dar mai ales mental. Mă face să descopăr idei, noi orizonturi și noi moduri de a gândi. Când călătoresc, mă simt ca un spirit liber și nu cred că există un sentiment mai bun decât acesta în lume.

CITITUL

Dragostea de lectură poate începe în mai multe moduri. Personal, mi-a plăcut mereu să citesc, încă de când eram copil. Întotdeauna am crezut despre cărți că au o formulă magică capabilă să te deschidă către lumi neașteptate și mereu am căutat răspunsuri în ele. Doar că, de-a lungul anilor, am înțeles despre cărți că nu îți dau răspunsuri, dar te deschid la numeroase întrebări, insinuează o mie de îndoieli, ușurează doar pentru o clipă anxietatea pe care curioșii o poartă înăuntrul





ANIMALELE

Pasiunea mea pentru animale a început când eram mică și, deși nu am avut niciodată un prieten blănos lângă mine, întotdeauna am găsit tandrețe în a-i vedea pe stăpâni și animalele lor plimbându-se fără grijii în aer liber.

Evident, nu mă refer doar la cele de companie, ci și la animalele sălbatice: am urmărit mereu cu admirație documentarele în care se vedeau lei răcnind, gazele alergând și maimuțe cățărându-se peste tot.
Cred că dragostea pe care ne-o dă un animal este necondiționată și de fiecare dată când văd animale maltratate mă doare inima. Din păcate, nu am cum să pun capăt acestor situații neplăcute, dar sunt sigur că oamenii, cu un pic de bunăvoință, își pot da seama de răul pe care îl fac și își pot remedia greșelile, din respect pentru animale și nu numai.

GOURMET

Întotdeauna am avut o slăbiciune pentru mâncare, dar pentru mâncarea făcută bine.

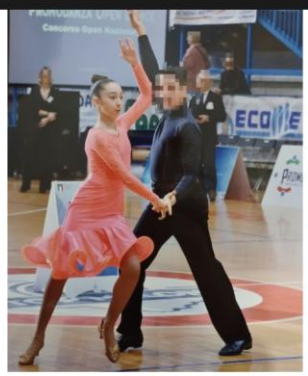
De când eram mică, mama m-a răsfățat cu mâncăruri delicioase făcute de ea și trebuie să spun că nu a fost cea mai bună alegere a ei, din moment ce nu mă pricep prea bine la gătit și, eventual, să încerc să-i recreez preparatele.

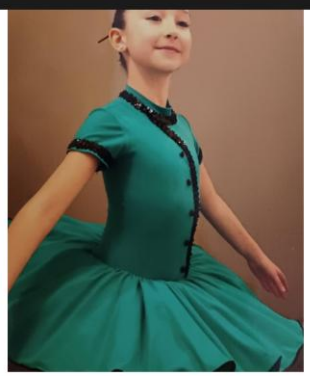
În ciuda faptului că sunt un dezastru în bucătărie, îmi place să încerc și să experimentez noi combinații de arome și gusturi.

După părerea mea, alimentele sărate sunt mult mai bune decât cele dulci, dar trebuie să spun că niciodată nu aș putea refuza o înghețată delicioasă cu fistic sau un baton de ciocolată!
Bucătăria mea preferată este cu siguranță cea italiană, cu aromele sale mediteraneene unice și inconfundabile, dar sunt foarte curioasă să încerc, pe viitor, mâncare din alte culturi, precum cea chineză sau cea spaniolă.

Jurnalul Erikăi

Pagină principală

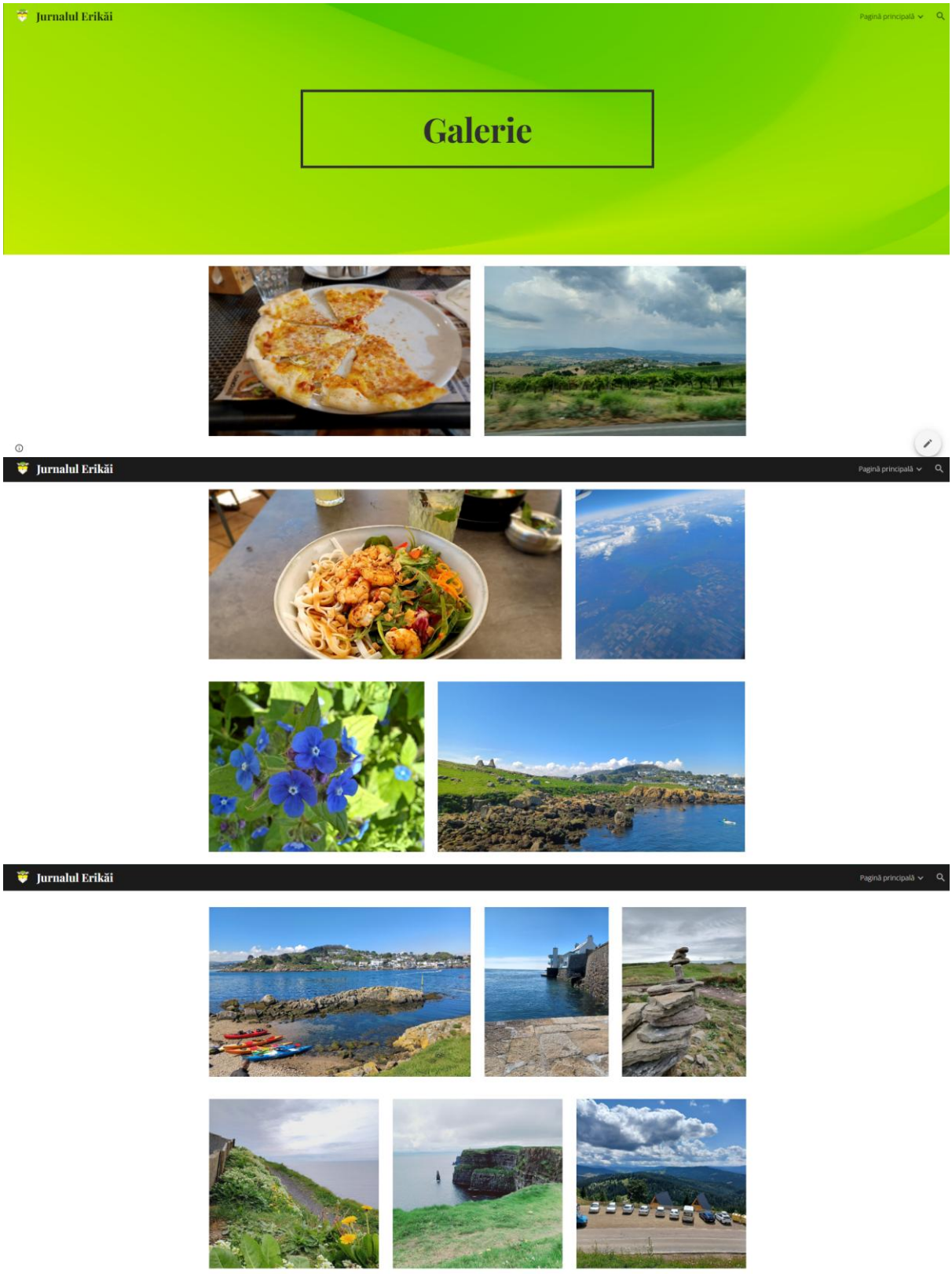




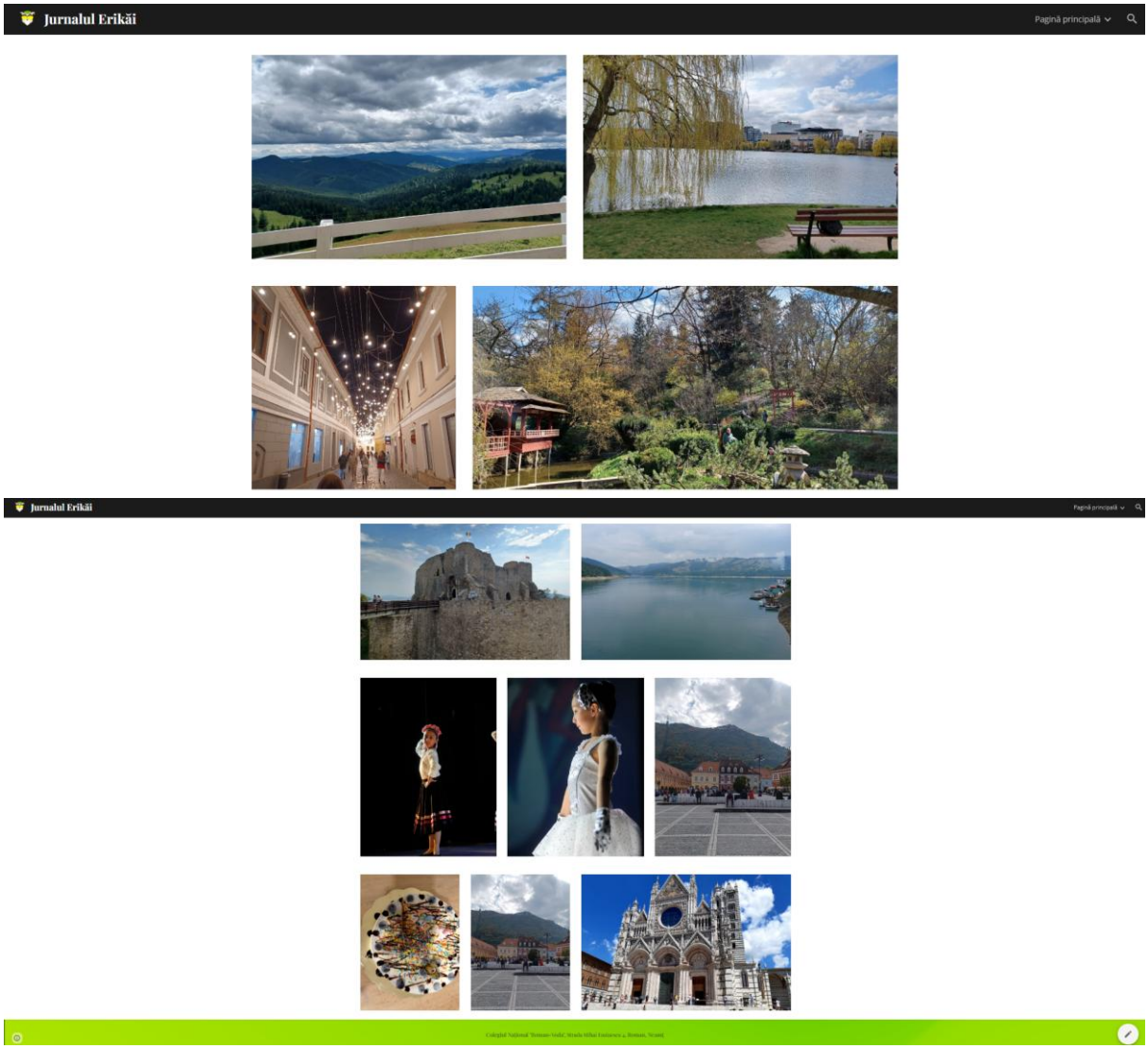
DANSUL

Dansul oferă cea mai mare libertate care poate exista: te face să te arăți fără fragilitate și teamă. Pasiunea este baza pentru dans. Ingredientele esențiale pentru atingerea scopului sunt angajamentul, determinarea, voința și curajul de a-și provoca limitele. Dar dansul este și disciplină, internă și externă, care reglează trupul și spiritul, este o școală de viață și disciplina este cea mai mare învățătură, pentru că ne face să înțelegem că visele se împlinesc doar cu perseverență și dăruire.
Din păcate, nu pot spune că am fost foarte norocoasă în acest domeniu: am schimbat mai multe școli de dans, parțial din cauza prejudecăților mele, parțial din cauza fricii de a fi judecată. Acest lucru mă întristează foarte tare, dar sunt sigură că deciziile pe care le-am luat în trecut au fost cele potrivite, deoarece cred în ele și știu că, chiar dacă dansul nu reprezintă viitorul pentru mine, va rămâne mereu o pasiune importantă. Îl voi vedea mereu ca pe un profesor care m-a învățat mult cu trecerea timpului.

Dobos Erika


Jurnalul Erikăi
Pagină principală
Galerie
Jurnalul Erikăi
Pagină principală
Jurnalul Erikăi
Pagină principală

Dobos Erika


Jurnalul Erikăi
Pagină principală
Jurnalul Erikăi
Pagină principală

Dobos Erika

Egalitate de sanse, toleranță și diversitate socială - proiect realizat în cadrul pregătirii culturale

EGALITATE DE SANSE, TOLERANTA SI DIVERSITATE SOCIALA



Powered by genially EDUCATION

CUPRINS

- INTRODUCERE SI IMPORTANTA
- EDUCATIA PENTRU EGALITATE
- INCLUZIUNEA SOCIALA
- STEREOTIPURI SI PREJUDECATI
- PREVENIREA DISCRIMINARII
- ORGANIZATII SI IMPORTANTA LOR



Powered by genially EDUCATION

Dobos Erika

INTRODUCERE

CE ESTE TOLERANȚA?

Toleranța este valoarea care stă la baza drepturilor universale ale omului și a libertăților fundamentale. Este vorba despre acceptarea și respectarea diferențelor dintre oameni, culturi, religii și opinii, fără discriminare sau prejudecăți.

CE ESTE EGALITATEA DE ȘANSE?

În general, termenul „egalitate de șanse" este utilizat pentru a se referi la condiția de paritate și egalitate introdusă pentru a garanta că toate persoanele sunt tratate în mod egal și pentru a preveni formele de discriminare pe baza anumitor aspecte (vârstă, etnie, handicap, orientare religioasă și politică etc.).

CE ESTE DIVERSITATEA SOCIALĂ?

Diversitatea socială este un proces prin care părți ale unei populații dobândesc treptat o identitate distinctă în ceea ce privește funcția, activitatea, structura, cultura, autoritatea, puterea sau alte caracteristici sociale semnificative și relevante.





> ***AM UN VIS CĂ ÎNTR-O BUNĂ ZI CEI PATRU COPII AI MEI VOR TRĂI ÎNTR-O NAȚIUNE ÎN CARE NU VOR MAI FI JUDECAȚI DUPĂ CULOAREA PIELII LOR, CI DUPĂ CARACTERUL LOR. MARTIN LUTHER KING***

Powered by genially EDUCATION

INCLUZIUNEA SOCIALĂ - FUNDAMENTUL EGALITĂȚII ÎN SOCIETATE

CE ESTE INCLUZIUNEA?

Incluziunea socială este un proces multidimensional prin care indivizii și grupurile marginalizate sunt integrate activ în viața economică, socială, culturală și politică a unei societăți. Conform cercetărilor din domeniul sociologiei și studiilor privind drepturile omului, incluziunea socială nu înseamnă doar absența discriminării, ci și crearea unor structuri și politici care asigură participarea echitabilă a tuturor cetățenilor.



FACTORII INCLUZIUNII SOCIALE

Incluziunea socială depinde de mai mulți factori esențiali, printre care:

- **Accesul la educație și formare profesională** – Oferă oportunități egale pentru dezvoltare și mobilitate socială.
- **Participarea economică** – Asigură integrarea pe piața muncii și reducerea decalajelor salariale.
- **Accesul la servicii publice și sănătate** – Contribuie la reducerea inegalităților și îmbunătățirea calității vieții.
- **Reprezentarea și participarea politică** – Garantează că toate grupurile au o voce în procesul decizional.

BENEFICIILE INCLUZIUNII SOCIALE

- **Coeziune socială** – Diminuează conflictele sociale și promovează solidaritatea.
- **Dezvoltare economică sustenabilă** – Studiile arată că societățile incluzive sunt mai prospere.
- **Creșterea capitalului social** – Relațiile de încredere și colaborare între diferite grupuri duc la o comunitate mai stabilă.

> ***DIVERSITATEA ESTE UN FAPT. INCLUZIUNEA ESTE O ALEGERE PE CARE O FACEM ZILNIC. EGALITATEA ESTE SCOPUL.***
>
> ***JOHNETTA BETSCH COLE***

Powered by genially EDUCATION

Dobos Erika

PREVENIREA DISCRIMINARII

Nimeni nu se naște urând o altă persoană din cauza culorii pielii, a trecutului sau sau a religiei sale. Oamenii învață să urască, atunci pot fi învățați să iubească.-Nelson Mandela

Prevenirea discriminării este un aspect esențial al unei societăți juste și echitabile. Aceasta presupune un set de acțiuni și strategii care vizează eliminarea inegalităților și asigurarea unui tratament egal pentru toți oamenii.



CÂTEVA MĂSURI PRIN CARE SE POATE PREVENI DISCRIMINAREA:

- **Educarea și sensibilizarea publicului**: Promovarea unor atitudini corecte și respectuoase față de diversitate prin campanii educaționale care explică conceptele de discriminare și diversitate.
- **Legea și reglementările anti-discriminare:** Crearea și implementarea unor legi clare care interzic discriminarea în toate formele sale, precum și sancționarea celor care încalcă aceste legi.
- **Promovarea incluziunii în locuri de muncă și educație**: Companiile și instituțiile educaționale pot implementa politici care garantează oportunități egale pentru toți

Powered by genially EDUCATION

EDUCATIA PENTRU EGALITATE

Educația pentru egalitate joacă un rol esențial în promovarea diversității și incluziunii, învățând elevii să recunoască și să combată stereotipurile și prejudecățile care există în societate. Prin educație, tinerii învață să aprecieze diferențele de rasă, religie, gen, orientare sexuală, dizabilitate sau orice altă caracteristică personală, dezvoltându-și astfel o înțelegere corectă a principiilor de egalitate și echitate.

„Educația este cheia pentru închiderea diferențelor de șanse și pentru crearea unei lumi mai egale.” – Malala Yousafzai



Acest proces educativ este crucial pentru crearea unui mediu social, economic și cultural în care toți indivizii, indiferent de background-ul lor, au șanse egale de a se dezvolta și de a-și atinge potențialul. Prin învățarea acestor valori, educația contribuie la formarea unor relații mai armonioase între grupuri sociale diferite, reducând discriminarea și creând o societate mai echitabilă.

Powered by genially EDUCATION

Dobos Erika

ACTIVITATI CARE INCURAJEAZA EDUCATIA PENTRU EGALITATE

Un exemplu practic al educației pentru egalitate ar putea fi implementarea unui program de sensibilizare asupra diversității în școli, cum ar fi "Lecții de Diversitate". Acesta poate include jocuri de rol, ateliere de discuție și întâlniri cu persoane din grupuri marginalizate, pentru a dezvolta empatia și respectul față de diferențele culturale, de gen sau religioase. În plus, bursele Erasmus pentru mobilitate studențească oferă o oportunitate reală de a învăța despre diversitate într-un context internațional, încurajând schimbul cultural și înțelegerea între tineri din diferite colțuri ale lumii. Aceste activități contribuie la formarea unei mentalități incluzive.

Powered by genially EDUCATION

STEREOTIPURI SI PREJUDECATI

În societatea modernă, diversitatea este o realitate de necontestat. Cu toate acestea, modul în care ne raportăm la cei diferiți de noi este adesea influențat de stereotipuri și prejudecăți. Aceste concepte, deși par inofensive la prima vedere, pot genera discriminare, excludere socială și conflicte.

CE ESTE UN STEREOTIP?

Un stereotip este o idee fixă despre un grup de oameni, fără să îi cunoaștem cu adevărat. Acestea se bazează pe trăsături cum ar fi genul, etnia, religia sau naționalitatea.

Spre exemplu:

- Fetele sunt mai sensibile și plâng ușor.
- Oamenii cu tatuaje sunt delincvenți sau rebeli.
- Fetele se pricep mai bine la gătit și la curățenie.
- Toți asiaticii sunt foarte inteligenți și buni la matematică.
- Oamenii bogați sunt aroganți și egoiști.

CE ESTE O PREJUDECATĂ?

O prejudecată este o opinie negativă formată înainte să cunoaștem o persoană, bazată doar pe aparențe sau apartenență. Acestea pot fi alimentate de familie, mass-media sau experiențe limitate.

Spre exemplu:

- Nu am încredere într-o femeie manager, sigur nu se va impune
- Dacă are accent străin, sigur nu știe limba bine.
- Dacă e sărac, sigur nu are bun-simț.
- E prea bătrân ca să mai învețe ceva nou.
- Are tatuaje, sigur e delincvent.

Powered by genially EDUCATION

Dobos Erika

STEREOTIPURI SI PREJUDECATI

CUM NE AFECTEAZĂ STEREOTIPURILE ȘI PREJUDECĂȚILE?

Efect	Exemplu concret
Excludere socială	Elevii care vin din altă țară sunt ignorați sau marginalizați.
Auto-limitare	Fetele evită specializări tehnice pentru că „nu sunt pentru ele”.
Discriminare	Candidații cu nume străine sunt respinși la angajare.
Creșterea tensiunilor sociale	Prejudecățile etnice pot duce la conflicte sau ură colectivă.

CUM COMBATEM STEREOTIPURILE ȘI PREJUDECĂȚILE?

Soluție	Exemplu practic
Educație pentru diversitate	Proiecte în școală despre culturi și comunități diverse.
Contact direct	Activități comune cu grupuri diferite (schimburi de experiență).
Promovarea modelelor pozitive	Personalități din grupuri diverse care au reușit.
Gândire critică	Analizăm informațiile și ne punem întrebări înainte să judecăm.
Implicare activă	Să luăm poziție când vedem discriminare în jurul nostru.

Powered by genially EDUCATION

ORGANIZATII CARE PROTEJEAZA MINORITATI

FRA (Agency for Fundamental Rights)
-este o organizație unică si independentă care se ocupă cu promovarea si protejarea drepturilor omului in UE

ONU(Organizația Națiunilor Unite)
-are printre obiective menținerea păcii mondiale și dezvoltarea unor relații de prietenie între țări

OSCE(Organization for Security and Co-operation in Europe)
-această organizație ajută FRA în a combata și dezvălui cazuri de violență din motive discriminatorii

Printre drepturile protejate se regăsesc:
-dreptul de a nu fi discriminat pe baza vârstei, dizabilităților sau religie
-dreptul de a avea datele personale protejate
-dreptul de a avea acces la justiție





Powered by genially EDUCATION

Dobos Erika

IMPORTANTA ORGANIZATIILOR



Protejarea și egalitatea minorităților

-indiferent de persoană ea trebuie să aibă drepturi și responsabilități egale, indiferent de loc

Asigurarea incluziunii minorităților

-doar pentru că ești diferit nu înseamnă că nu meriți să fi inclus

Menținerea identității sociale ale individului

-orice persoană are dreptul să își exprime liber orientările și mentalitățile

Minimalizarea discriminării între persoane

-o lume în care fiecare persoană este acceptată așa cum este ea este idealul spre care ar trebui să îl atingem



Powered by genially EDUCATION

IMPORTANTA TEMEI IN CADRUL PROIECTULUI

Creșterea conștientizării asupra drepturilor fundamentale

Această conștientizare stimulează respectarea drepturilor fiecărei persoane, indiferent de context.

Promovarea unei mentalități incluzive

Cunoașterea acestei teme ne ajută să identificam situațiile de nedreptate și să susținem incluziunea și egalitatea.

Dezvoltarea empatiei și toleranței

Aceste calități sunt esențiale pentru relații armonioase și colaborare internațională.

Înțelegerea și adaptarea într-un mediu cultural diferit

Ne ajută să fim mai deschiși și mai respectuoși față de diversitatea culturală și socială, facilitând integrarea într-un mediu nou.



Powered by genially EDUCATION

BIBLIOGRAFIE

- *https://fra.europa.eu/ro*
- *https://epochtimes-romania.com/news/toleranta-o-valoare-necesara-convietuirii-pasnice---320667*
- *https://www.britishcouncil.ro/despre-noi/sanse-egale-diversitate#:~:text=Egalitatea%20de%20%C8%99anse%20%C3%AEnseamn%C4%83%20a,de%20lucru%20%C8%99i%20%C3%AEn%20societate.*
- *https://gep.unibuc.ro/wp-content/uploads/2023/01/Modul-Diversitate-egalitate-si-incluziune-sociala%CC%86.pdf*
- *https://eige.europa.eu/gender-based-violence*
- *https://www.europarl.europa.eu/about-parliament/ro/democracy-and-human-rights/fundamental-rights-in-the-eu/promoting-equal-opportunities*

Powered by genially EDUCATION

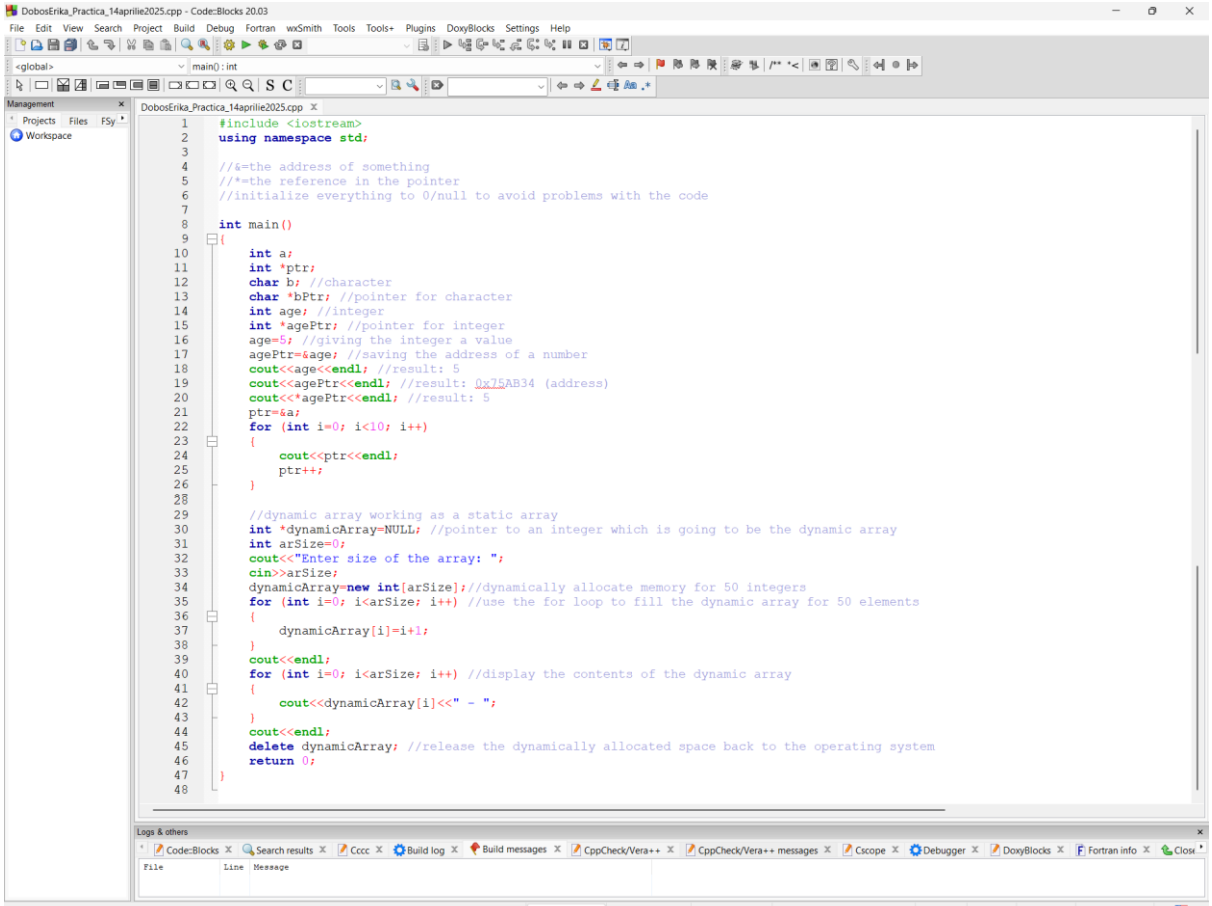
MULTUMIM PENTRU ATENTIE!

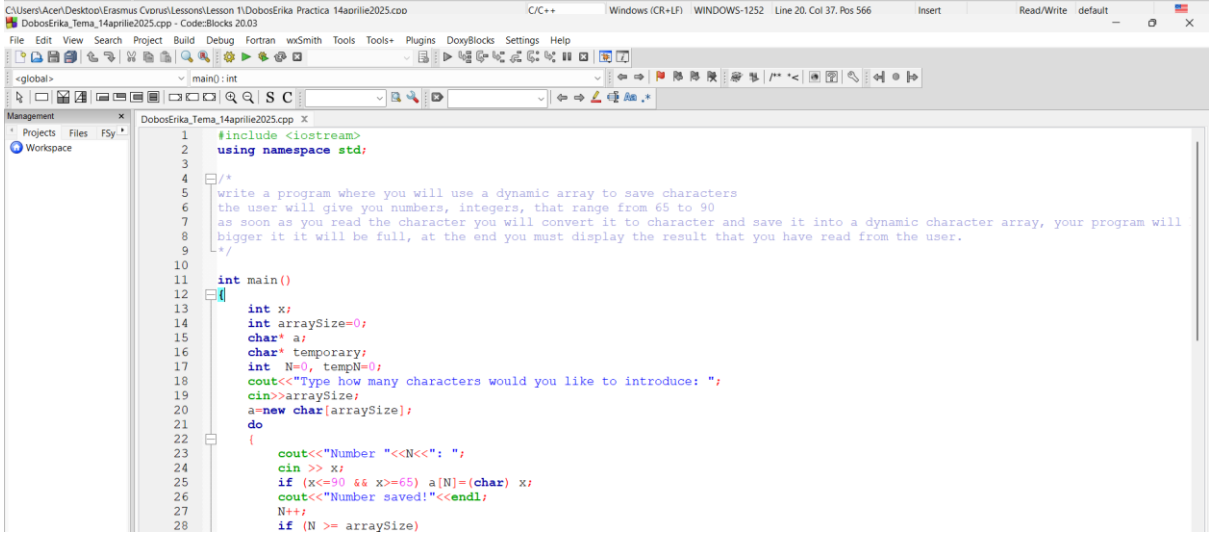


Powered by genially EDUCATION

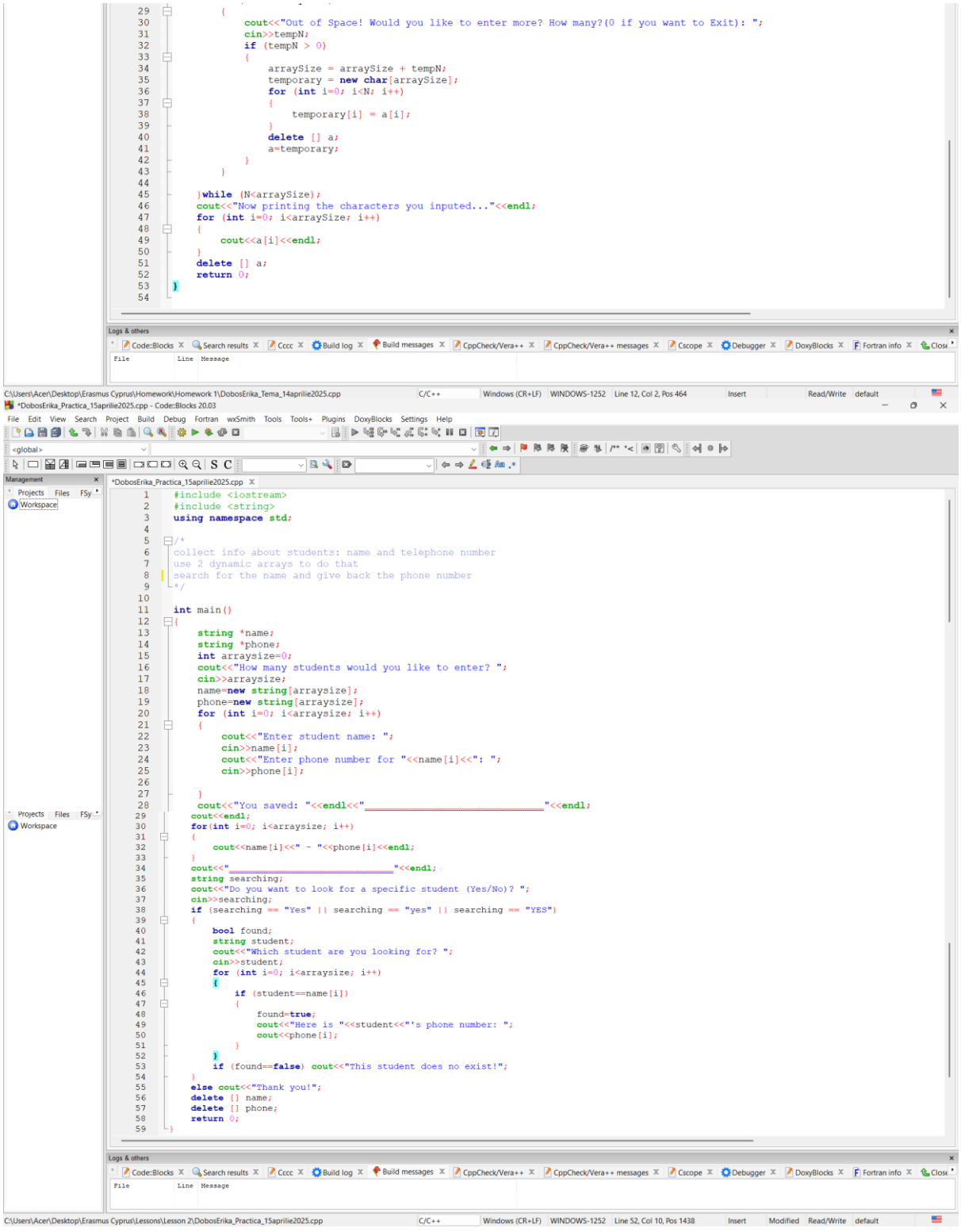
Dobos Erika

Algoritmi C++





Dobos Erika



```
        {
            cout<<"Out of Space! Would you like to enter more? How many?(0 if you want to Exit): ";
            cin>>tempN;
            if (tempN > 0)
            {
                arraySize = arraySize + tempN;
                temporary = new char[arraySize];
                for (int i=0; i<N; i++)
                {
                    temporary[i] = a[i];
                }
                delete [] a;
                a=temporary;
            }
        }

    }while (N<arraySize);
    cout<<"Now printing the characters you inputed..."<<endl;
    for (int i=0; i<arraySize; i++)
    {
        cout<<a[i]<<endl;
    }
    delete [] a;
    return 0;
}
```

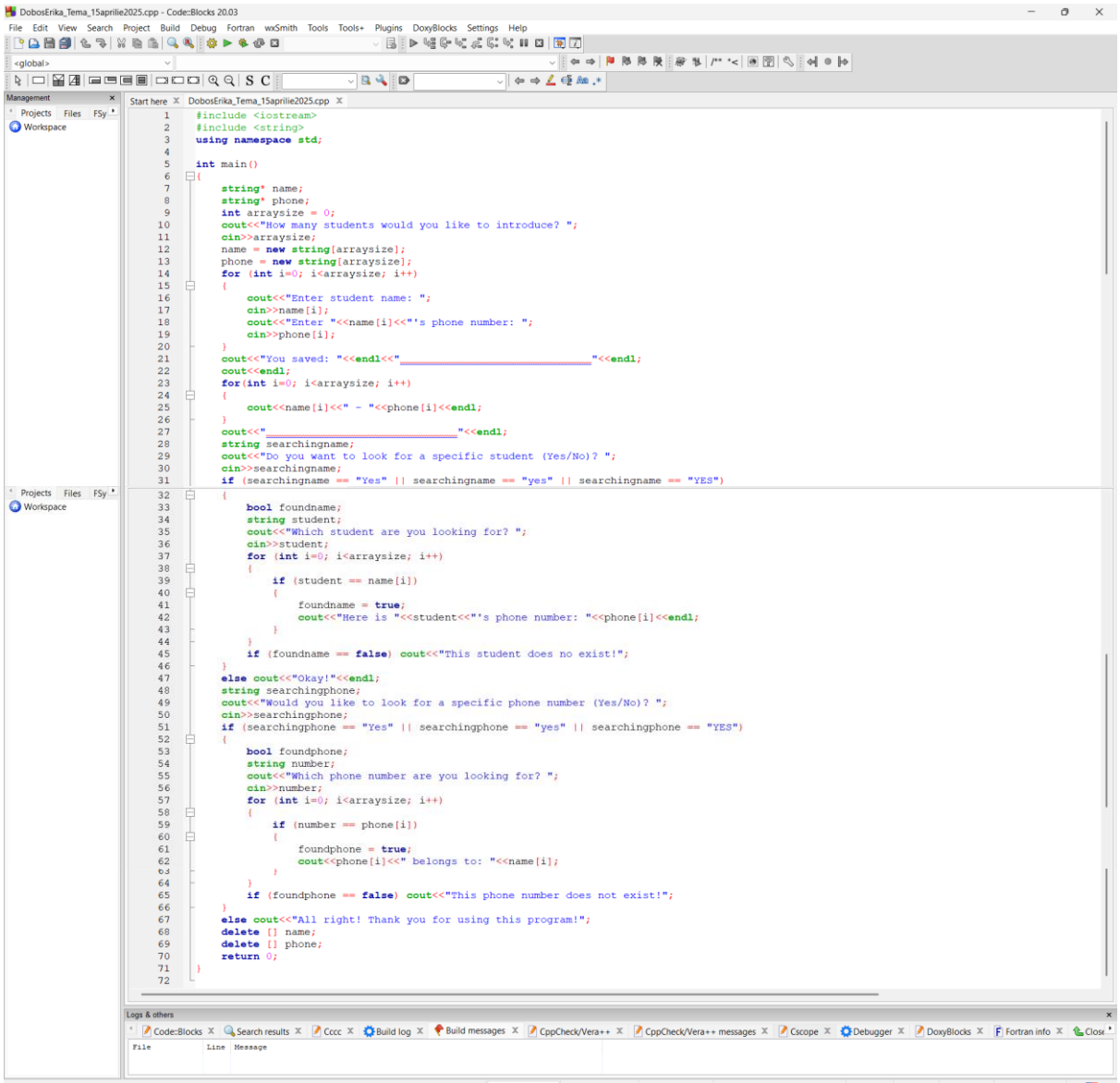
```
#include <iostream>
#include <string>
using namespace std;

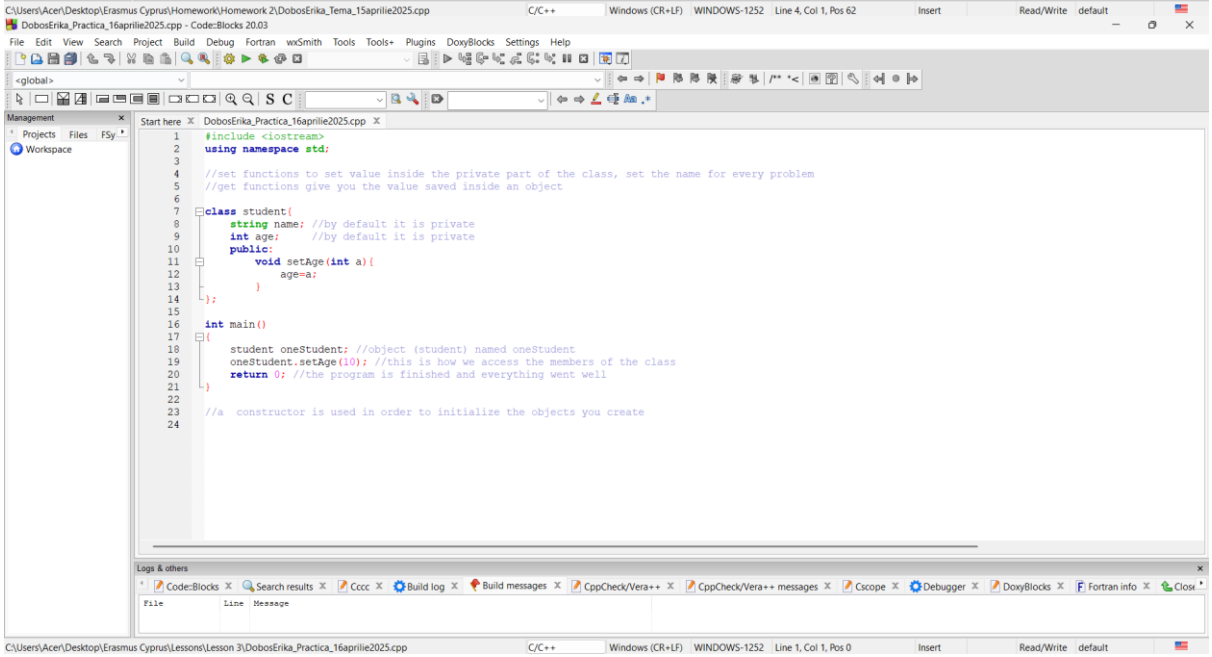
/*
collect info about students: name and telephone number
use 2 dynamic arrays to do that
search for the name and give back the phone number
*/

int main()
{
    string *name;
    string *phone;
    int arraysize=0;
    cout<<"How many students would you like to enter? ";
    cin>>arraysize;
    name=new string[arraysize];
    phone=new string[arraysize];
    for (int i=0; i<arraysize; i++)
    {
        cout<<"Enter student name: ";
        cin>>name[i];
        cout<<"Enter phone number for "<<name[i]<<": ";
        cin>>phone[i];

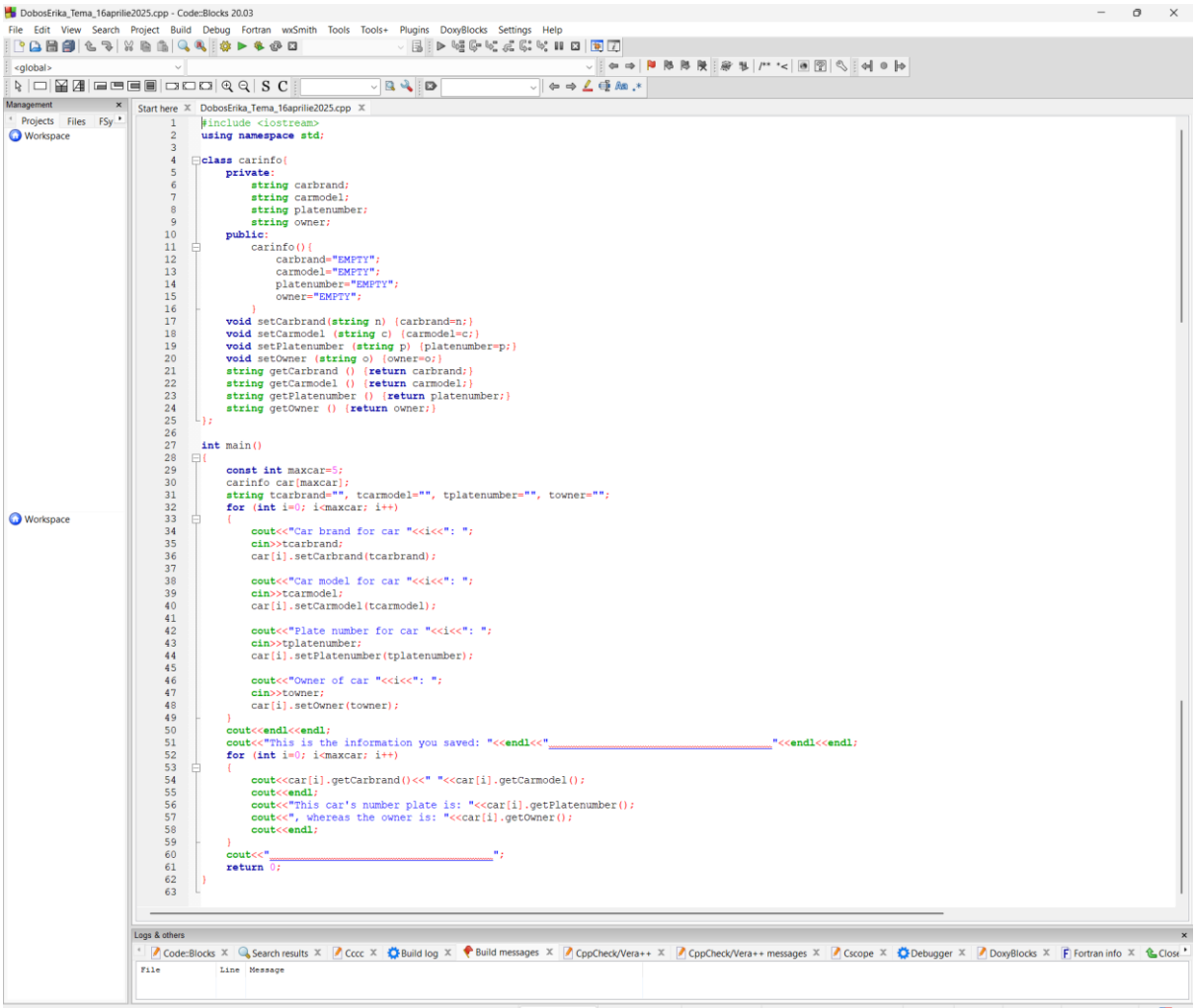
    }
    cout<<"You saved: "<<endl<<"_____"<<endl;
    cout<<endl;
    for(int i=0; i<arraysize; i++)
    {
        cout<<name[i]<<" - "<<phone[i]<<endl;
    }
    cout<<"_____"<<endl;
    string searching;
    cout<<"Do you want to look for a specific student (Yes/No)? ";
    cin>>searching;
    if (searching == "Yes" || searching == "yes" || searching == "YES")
    {
        bool found;
        string student;
        cout<<"Which student are you looking for? ";
        cin>>student;
        for (int i=0; i<arraysize; i++)
        {
            if (student==name[i])
            {
                found=true;
                cout<<"Here is "<<student<<"'s phone number: ";
                cout<<phone[i];
            }
        }
        if (found==false) cout<<"This student does no exist!";
    }
    else cout<<"Thank you!";
    delete [] name;
    delete [] phone;
    return 0;
}
```

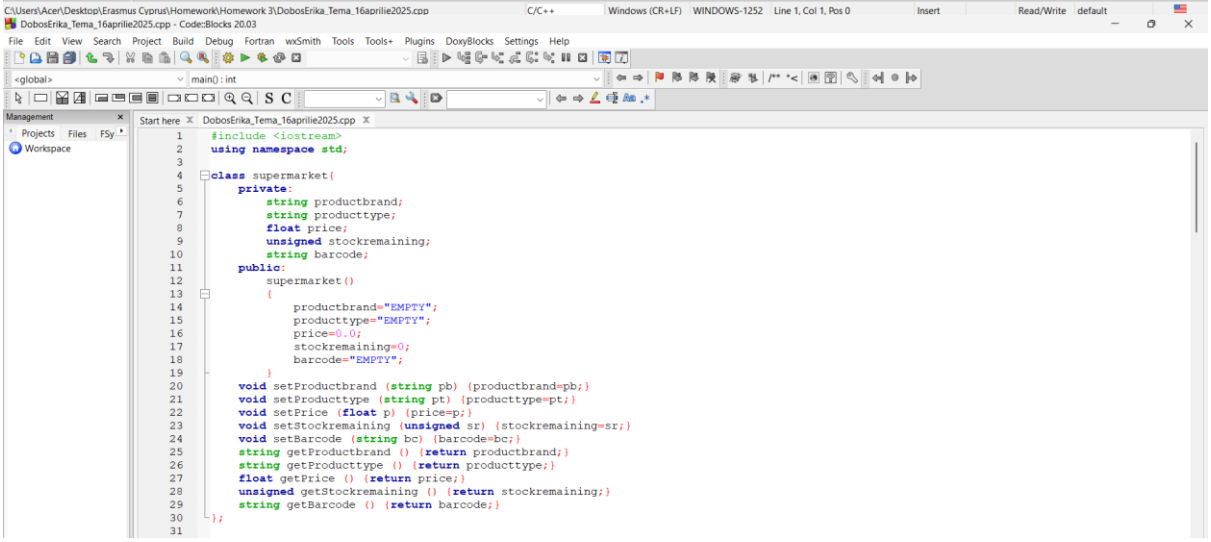
Dobos Erika


DobosErika_Tema_15aprilie2025.cpp - Code::Blocks 20.03
#include <iostream>
#include <string>
using namespace std;
int main()
{
string* name;
string* phone;
int arraysize = 0;
cout<<"How many students would you like to introduce? ";
cin>>arraysize;
name = new string[arraysize];
phone = new string[arraysize];
for (int i=0; i<arraysize; i++)
{
cout<<"Enter student name: ";
cin>>name[i];
cout<<"Enter "<<name[i]<<"'s phone number: ";
cin>>phone[i];
}
cout<<"You saved: "<<endl<<"_____"<<endl;
cout<<endl;
for(int i=0; i<arraysize; i++)
{
cout<<name[i]<<" - "<<phone[i]<<endl;
}
cout<<"_____"<<endl;
string searchingname;
cout<<"Do you want to look for a specific student (Yes/No)? ";
cin>>searchingname;
if (searchingname == "Yes" || searchingname == "yes" || searchingname == "YES")
{
bool foundname;
string student;
cout<<"Which student are you looking for? ";
cin>>student;
for (int i=0; i<arraysize; i++)
{
if (student == name[i])
{
foundname = true;
cout<<"Here is "<<student<<"'s phone number: "<<phone[i]<<endl;
}
}
if (foundname == false) cout<<"This student does no exist!";
}
else cout<<"Okay!"<<endl;
string searchingphone;
cout<<"Would you like to look for a specific phone number (Yes/No)? ";
cin>>searchingphone;
if (searchingphone == "Yes" || searchingphone == "yes" || searchingphone == "YES")
{
bool foundphone;
string number;
cout<<"Which phone number are you looking for? ";
cin>>number;
for (int i=0; i<arraysize; i++)
{
if (number == phone[i])
{
foundphone = true;
cout<<phone[i]<<" belongs to: "<<name[i];
}
}
if (foundphone == false) cout<<"This phone number does not exist!";
}
else cout<<"All right! Thank you for using this program!";
delete [] name;
delete [] phone;
return 0;
}

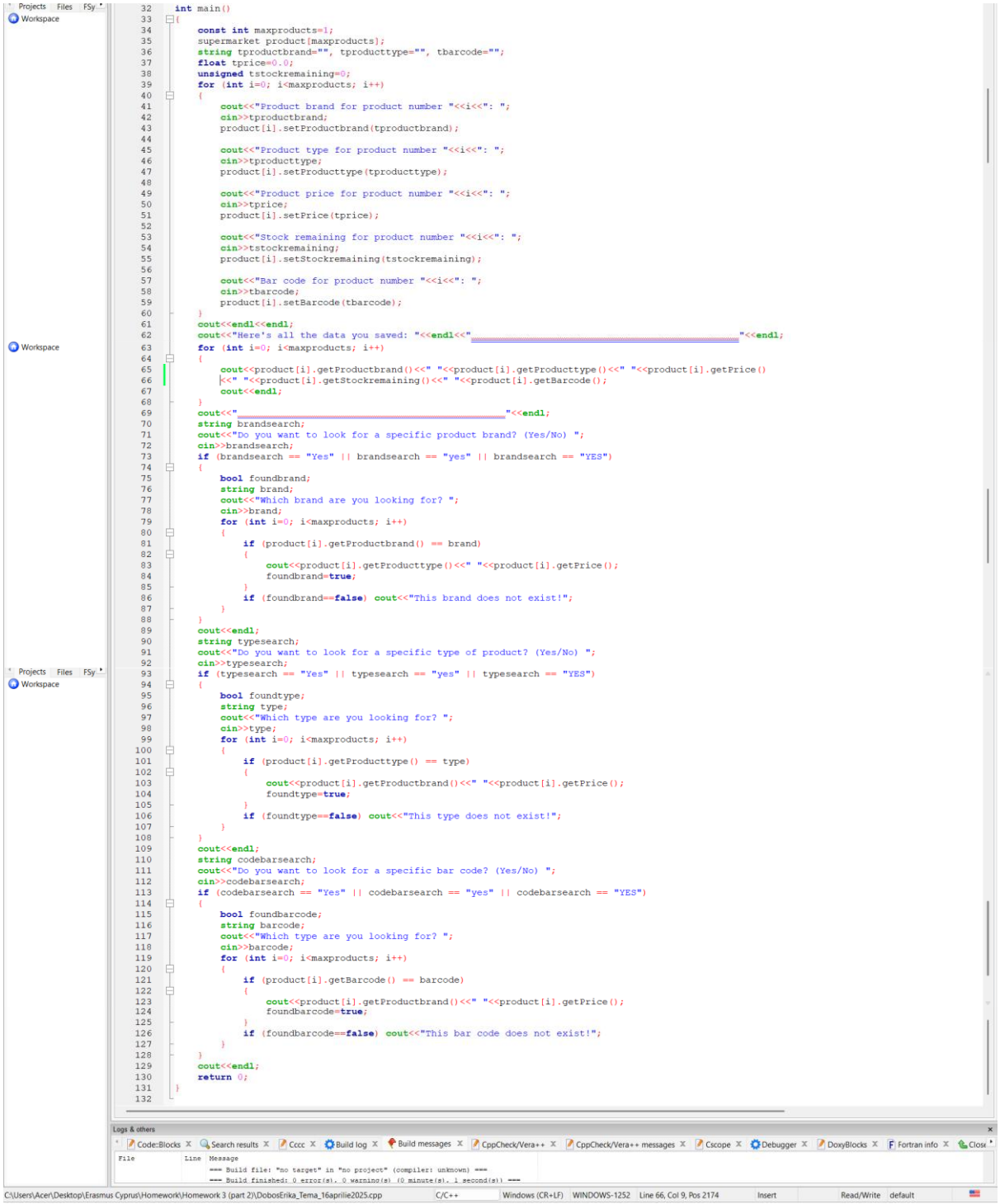

DobosErika_Practica_16aprilie2025.cpp - Code::Blocks 20.03
#include <iostream>
using namespace std;
//set functions to set value inside the private part of the class, set the name for every problem
//get functions give you the value saved inside an object
class student{
string name; //by default it is private
int age; //by default it is private
public:
void setAge(int a){
age=a;
}
};
int main()
{
student oneStudent; //object (student) named oneStudent
oneStudent.setAge(10); //this is how we access the members of the class
return 0; //the program is finished and everything went well
}
//a constructor is used in order to initialize the objects you create

Dobos Erika


DobosErika_Tema_16aprilie2025.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Files FSy
Workspace
Start here
DobosErika_Tema_16aprilie2025.cpp
#include <iostream>
using namespace std;
class carinfo{
private:
string carbrand;
string carmodel;
string platenumber;
string owner;
public:
carinfo(){
carbrand="EMPTY";
carmodel="EMPTY";
platenumber="EMPTY";
owner="EMPTY";
}
void setCarbrand(string n) {carbrand=n;}
void setCarmodel (string c) {carmodel=c;}
void setPlatenumber (string p) {platenumber=p;}
void setOwner (string o) {owner=o;}
string getCarbrand () {return carbrand;}
string getCarmodel () {return carmodel;}
string getPlatenumber () {return platenumber;}
string getOwner () {return owner;}
};
int main()
{
const int maxcar=5;
carinfo car[maxcar];
string tcarbrand="", tcarmodel="", tplatenumber="", towner="";
for (int i=0; i<maxcar; i++)
{
cout<<"Car brand for car "<<i<<": ";
cin>>tcarbrand;
car[i].setCarbrand(tcarbrand);
cout<<"Car model for car "<<i<<": ";
cin>>tcarmodel;
car[i].setCarmodel(tcarmodel);
cout<<"Plate number for car "<<i<<": ";
cin>>tplatenumber;
car[i].setPlatenumber(tplatenumber);
cout<<"Owner of car "<<i<<": ";
cin>>towner;
car[i].setOwner(towner);
}
cout<<endl<<endl;
cout<<"This is the information you saved: "<<endl<<"_____"<<endl<<endl;
for (int i=0; i<maxcar; i++)
{
cout<<car[i].getCarbrand()<<" "<<car[i].getCarmodel();
cout<<endl;
cout<<"This car's number plate is: "<<car[i].getPlatenumber();
cout<<", whereas the owner is: "<<car[i].getOwner();
cout<<endl;
}
cout<<"_____";
return 0;
}
Logs & others
Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger DoxyBlocks Fortran info
File Line Message
C:\Users\Acer\Desktop\Erasmus Cyprus\Homework\Homework 3\DobosErika_Tema_16aprilie2025.cpp
C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert Read/Write default


DobosErika_Tema_16aprilie2025.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main() : int
Management
Projects Files FSy
Workspace
Start here
DobosErika_Tema_16aprilie2025.cpp
#include <iostream>
using namespace std;
class supermarket{
private:
string productbrand;
string producttype;
float price;
unsigned stockremaining;
string barcode;
public:
supermarket()
{
productbrand="EMPTY";
producttype="EMPTY";
price=0.0;
stockremaining=0;
barcode="EMPTY";
}
void setProductbrand (string pb) {productbrand=pb;}
void setProducttype (string pt) {producttype=pt;}
void setPrice (float p) {price=p;}
void setStockremaining (unsigned sr) {stockremaining=sr;}
void setBarcode (string bc) {barcode=bc;}
string getProductbrand () {return productbrand;}
string getProducttype () {return producttype;}
float getPrice () {return price;}
unsigned getStockremaining () {return stockremaining;}
string getBarcode () {return barcode;}
};

Dobos Erika



```cpp
int main()
{
    const int maxproducts=1;
    supermarket product[maxproducts];
    string tproductbrand="", tproducttype="", tbarcode="";
    float tprice=0.0;
    unsigned tstockremaining=0;
    for (int i=0; i<maxproducts; i++)
    {
        cout<<"Product brand for product number "<<i<<": ";
        cin>>tproductbrand;
        product[i].setProductbrand(tproductbrand);

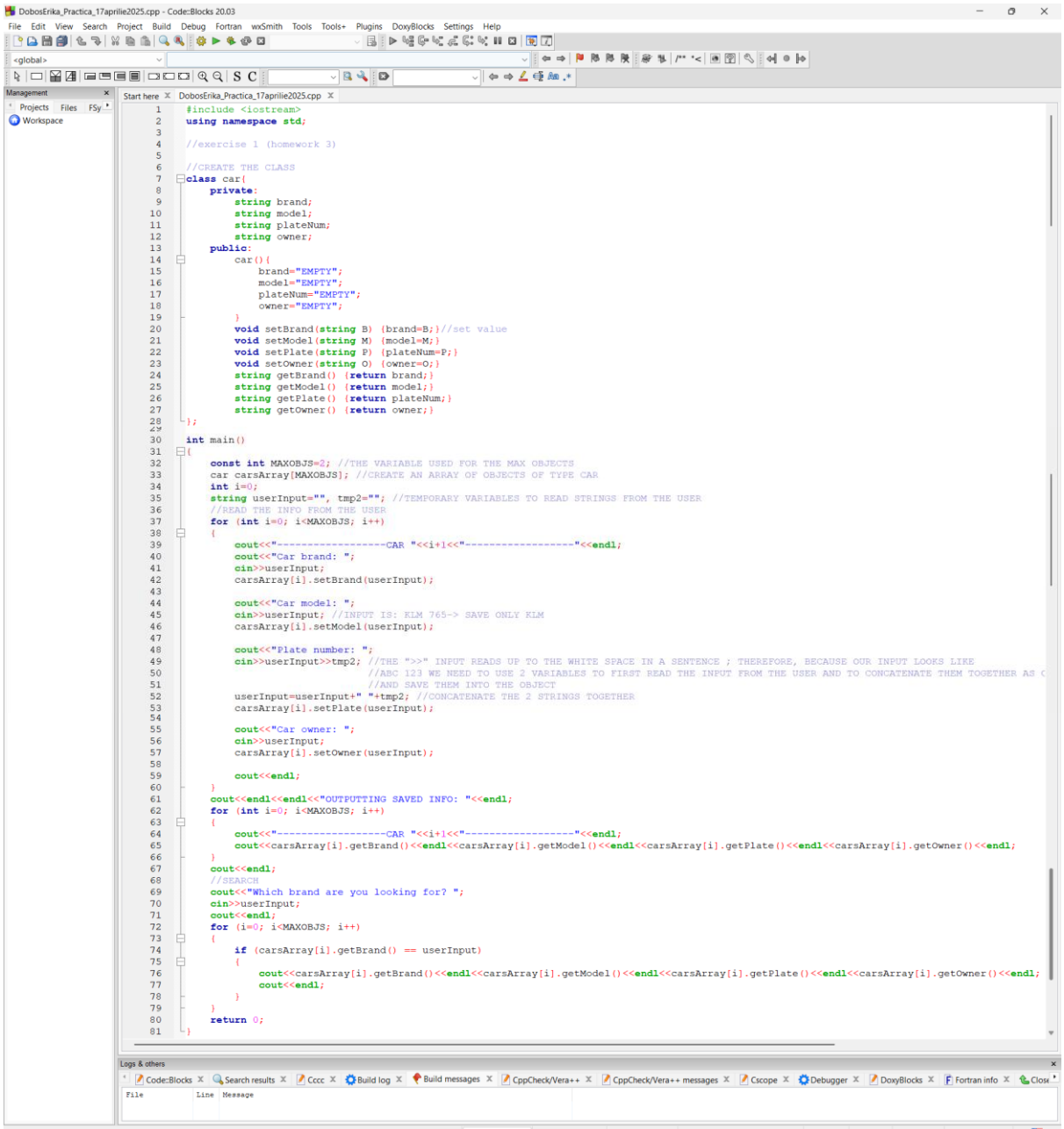
        cout<<"Product type for product number "<<i<<": ";
        cin>>tproducttype;
        product[i].setProducttype(tproducttype);

        cout<<"Product price for product number "<<i<<": ";
        cin>>tprice;
        product[i].setPrice(tprice);

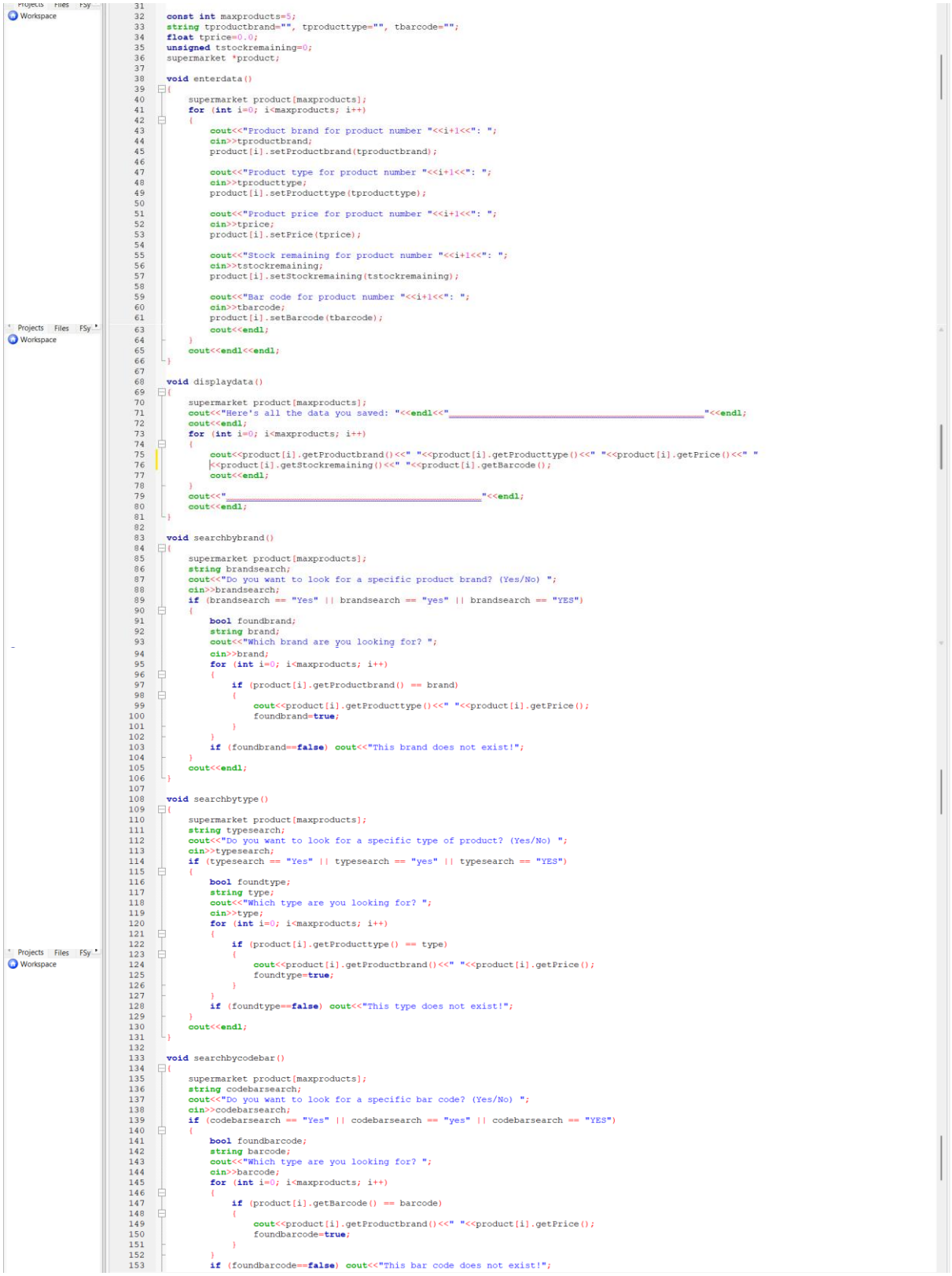
        cout<<"Stock remaining for product number "<<i<<": ";
        cin>>tstockremaining;
        product[i].setStockremaining(tstockremaining);

        cout<<"Bar code for product number "<<i<<": ";
        cin>>tbarcode;
        product[i].setBarcode(tbarcode);
    }
    cout<<endl<<endl;
    cout<<"Here's all the data you saved: "<<endl<<"_____"<<endl;
    for (int i=0; i<maxproducts; i++)
    {
        cout<<product[i].getProductbrand()<<" "<<product[i].getProducttype()<<" "<<product[i].getPrice()
        <<" "<<product[i].getStockremaining()<<" "<<product[i].getBarcode();
        cout<<endl;
    }
    cout<<"_____"<<endl;
    string brandsearch;
    cout<<"Do you want to look for a specific product brand? (Yes/No) ";
    cin>>brandsearch;
    if (brandsearch == "Yes" || brandsearch == "yes" || brandsearch == "YES")
    {
        bool foundbrand;
        string brand;
        cout<<"Which brand are you looking for? ";
        cin>>brand;
        for (int i=0; i<maxproducts; i++)
        {
            if (product[i].getProductbrand() == brand)
            {
                cout<<product[i].getProducttype()<<" "<<product[i].getPrice();
                foundbrand=true;
            }
            if (foundbrand==false) cout<<"This brand does not exist!";
        }
    }
    cout<<endl;
    string typesearch;
    cout<<"Do you want to look for a specific type of product? (Yes/No) ";
    cin>>typesearch;
    if (typesearch == "Yes" || typesearch == "yes" || typesearch == "YES")
    {
        bool foundtype;
        string type;
        cout<<"Which type are you looking for? ";
        cin>>type;
        for (int i=0; i<maxproducts; i++)
        {
            if (product[i].getProducttype() == type)
            {
                cout<<product[i].getProductbrand()<<" "<<product[i].getPrice();
                foundtype=true;
            }
            if (foundtype==false) cout<<"This type does not exist!";
        }
    }
    cout<<endl;
    string codebarsearch;
    cout<<"Do you want to look for a specific bar code? (Yes/No) ";
    cin>>codebarsearch;
    if (codebarsearch == "Yes" || codebarsearch == "yes" || codebarsearch == "YES")
    {
        bool foundbarcode;
        string barcode;
        cout<<"Which type are you looking for? ";
        cin>>barcode;
        for (int i=0; i<maxproducts; i++)
        {
            if (product[i].getBarcode() == barcode)
            {
                cout<<product[i].getProductbrand()<<" "<<product[i].getPrice();
                foundbarcode=true;
            }
            if (foundbarcode==false) cout<<"This bar code does not exist!";
        }
    }
    cout<<endl;
    return 0;
}
```

Dobos Erika

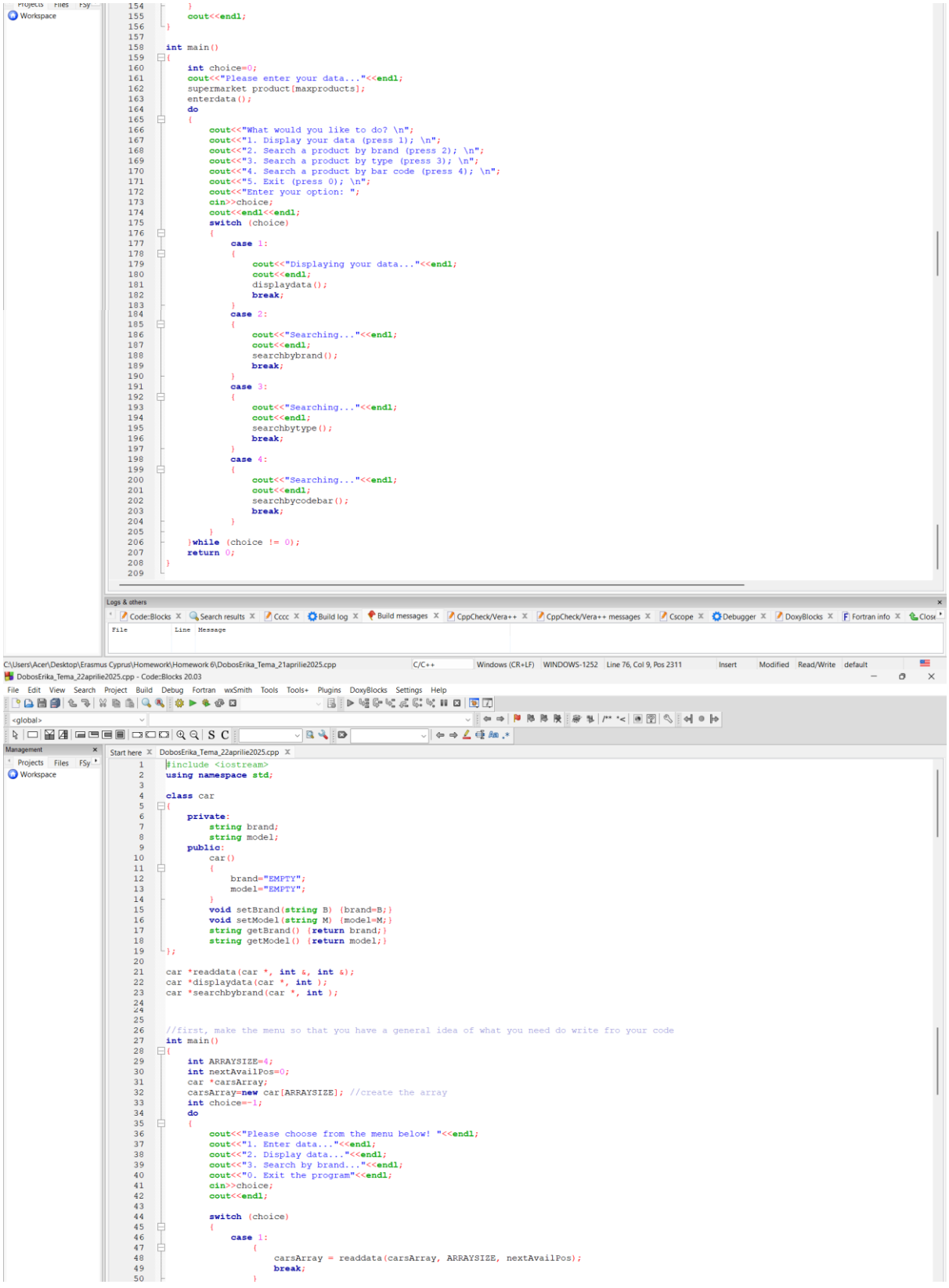

DobosErika_Practica_17aprilie2025.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Files FSy
Workspace
Start here
DobosErika_Practica_17aprilie2025.cpp
#include <iostream>
using namespace std;
//exercise 1 (homework 3)
//CREATE THE CLASS
class car{
private:
string brand;
string model;
string plateNum;
string owner;
public:
car(){
brand="EMPTY";
model="EMPTY";
plateNum="EMPTY";
owner="EMPTY";
}
void setBrand(string B) {brand=B;}//set value
void setModel(string M) {model=M;}
void setPlate(string P) {plateNum=P;}
void setOwner(string O) {owner=O;}
string getBrand() {return brand;}
string getModel() {return model;}
string getPlate() {return plateNum;}
string getOwner() {return owner;}
};
int main()
{
const int MAXOBJS=2; //THE VARIABLE USED FOR THE MAX OBJECTS
car carsArray[MAXOBJS]; //CREATE AN ARRAY OF OBJECTS OF TYPE CAR
int i=0;
string userInput="", tmp2=""; //TEMPORARY VARIABLES TO READ STRINGS FROM THE USER
//READ THE INFO FROM THE USER
for (int i=0; i<MAXOBJS; i++)
{
cout<<"------------------CAR "<<i+1<<"------------------"<<endl;
cout<<"Car brand: ";
cin>>userInput;
carsArray[i].setBrand(userInput);
cout<<"Car model: ";
cin>>userInput; //INPUT IS: KLM 765-> SAVE ONLY KLM
carsArray[i].setModel(userInput);
cout<<"Plate number: ";
cin>>userInput>>tmp2; //THE ">>" INPUT READS UP TO THE WHITE SPACE IN A SENTENCE ; THEREFORE, BECAUSE OUR INPUT LOOKS LIKE
//ABC 123 WE NEED TO USE 2 VARIABLES TO FIRST READ THE INPUT FROM THE USER AND TO CONCATENATE THEM TOGETHER AS
//AND SAVE THEM INTO THE OBJECT
userInput=userInput+" "+tmp2; //CONCATENATE THE 2 STRINGS TOGETHER
carsArray[i].setPlate(userInput);
cout<<"Car owner: ";
cin>>userInput;
carsArray[i].setOwner(userInput);
cout<<endl;
}
cout<<endl<<endl<<"OUTPUTTING SAVED INFO: "<<endl;
for (int i=0; i<MAXOBJS; i++)
{
cout<<"------------------CAR "<<i+1<<"------------------"<<endl;
cout<<carsArray[i].getBrand()<<endl<<carsArray[i].getModel()<<endl<<carsArray[i].getPlate()<<endl<<carsArray[i].getOwner()<<endl;
}
cout<<endl;
//SEARCH
cout<<"Which brand are you looking for? ";
cin>>userInput;
cout<<endl;
for (i=0; i<MAXOBJS; i++)
{
if (carsArray[i].getBrand() == userInput)
{
cout<<carsArray[i].getBrand()<<endl<<carsArray[i].getModel()<<endl<<carsArray[i].getPlate()<<endl<<carsArray[i].getOwner()<<endl;
cout<<endl;
}
}
return 0;
}
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck/Vera++
CppCheck/Vera++ messages
Cscope
Debugger
DoxyBlocks
Fortran info
File
Line
Message
C:\Users\Acer\Desktop\Erasmus Cyprus\Lessons\Lesson 4\DobosErika_Practica_17aprilie2025.cpp
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 1, Col 1, Pos 0
Insert
Read/Write
default
*DobosErika_Tema_21aprilie2025.cpp - Code::Blocks 20.03
displaydata() : void
*DobosErika_Tema_21aprilie2025.cpp
#include <iostream>
using namespace std;
class supermarket{
private:
string productbrand;
string producttype;
float price;
unsigned stockremaining;
string barcode;
public:
supermarket()
{
productbrand="EMPTY";
producttype="EMPTY";
price=0.0;
stockremaining=0;
barcode="EMPTY";
}
void setProductbrand (string pb) {productbrand=pb;}
void setProducttype (string pt) {producttype=pt;}
void setPrice (float p) {price=p;}
void setStockremaining (unsigned sr) {stockremaining=sr;}
void setBarcode (string bc) {barcode=bc;}
string getProductbrand () {return productbrand;}
string getProducttype () {return producttype;}
float getPrice () {return price;}
unsigned getStockremaining () {return stockremaining;}
string getBarcode () {return barcode;}
};

Dobos Erika



```
31
32  const int maxproducts=5;
33  string tproductbrand="", tproducttype="", tbarcode="";
34  float tprice=0.0;
35  unsigned tstockremaining=0;
36  supermarket *product;
37
38  void enterdata()
39  {
40      supermarket product[maxproducts];
41      for (int i=0; i<maxproducts; i++)
42      {
43          cout<<"Product brand for product number "<<i+1<<": ";
44          cin>>tproductbrand;
45          product[i].setProductbrand(tproductbrand);
46
47          cout<<"Product type for product number "<<i+1<<": ";
48          cin>>tproducttype;
49          product[i].setProducttype(tproducttype);
50
51          cout<<"Product price for product number "<<i+1<<": ";
52          cin>>tprice;
53          product[i].setPrice(tprice);
54
55          cout<<"Stock remaining for product number "<<i+1<<": ";
56          cin>>tstockremaining;
57          product[i].setStockremaining(tstockremaining);
58
59          cout<<"Bar code for product number "<<i+1<<": ";
60          cin>>tbarcode;
61          product[i].setBarcode(tbarcode);
63          cout<<endl;
64      }
65      cout<<endl<<endl;
66  }
67
68  void displaydata()
69  {
70      supermarket product[maxproducts];
71      cout<<"Here's all the data you saved: "<<endl<<"_____"<<endl;
72      cout<<endl;
73      for (int i=0; i<maxproducts; i++)
74      {
75          cout<<product[i].getProductbrand()<<" "<<product[i].getProducttype()<<" "<<product[i].getPrice()<<" "
76          <<product[i].getStockremaining()<<" "<<product[i].getBarcode();
77          cout<<endl;
78      }
79      cout<<"_____"<<endl;
80      cout<<endl;
81  }
82
83  void searchbybrand()
84  {
85      supermarket product[maxproducts];
86      string brandsearch;
87      cout<<"Do you want to look for a specific product brand? (Yes/No) ";
88      cin>>brandsearch;
89      if (brandsearch == "Yes" || brandsearch == "yes" || brandsearch == "YES")
90      {
91          bool foundbrand;
92          string brand;
93          cout<<"Which brand are you looking for? ";
94          cin>>brand;
95          for (int i=0; i<maxproducts; i++)
96          {
97              if (product[i].getProductbrand() == brand)
98              {
99                  cout<<product[i].getProducttype()<<" "<<product[i].getPrice();
100                 foundbrand=true;
101             }
102         }
103         if (foundbrand==false) cout<<"This brand does not exist!";
104     }
105     cout<<endl;
106 }
107
108 void searchbytype()
109 {
110     supermarket product[maxproducts];
111     string typesearch;
112     cout<<"Do you want to look for a specific type of product? (Yes/No) ";
113     cin>>typesearch;
114     if (typesearch == "Yes" || typesearch == "yes" || typesearch == "YES")
115     {
116         bool foundtype;
117         string type;
118         cout<<"Which type are you looking for? ";
119         cin>>type;
120         for (int i=0; i<maxproducts; i++)
121         {
122             if (product[i].getProducttype() == type)
123             {
124                 cout<<product[i].getProductbrand()<<" "<<product[i].getPrice();
125                 foundtype=true;
126             }
127         }
128         if (foundtype==false) cout<<"This type does not exist!";
129     }
130     cout<<endl;
131 }
132
133 void searchbycodebar()
134 {
135     supermarket product[maxproducts];
136     string codebarsearch;
137     cout<<"Do you want to look for a specific bar code? (Yes/No) ";
138     cin>>codebarsearch;
139     if (codebarsearch == "Yes" || codebarsearch == "yes" || codebarsearch == "YES")
140     {
141         bool foundbarcode;
142         string barcode;
143         cout<<"Which type are you looking for? ";
144         cin>>barcode;
145         for (int i=0; i<maxproducts; i++)
146         {
147             if (product[i].getBarcode() == barcode)
148             {
149                 cout<<product[i].getProductbrand()<<" "<<product[i].getPrice();
150                 foundbarcode=true;
151             }
152         }
153         if (foundbarcode==false) cout<<"This bar code does not exist!";
```

Dobos Erika



```
    }
    cout<<endl;
}

int main()
{
    int choice=0;
    cout<<"Please enter your data..."<<endl;
    supermarket product[maxproducts];
    enterdata();
    do
    {
        cout<<"What would you like to do? \n";
        cout<<"1. Display your data (press 1); \n";
        cout<<"2. Search a product by brand (press 2); \n";
        cout<<"3. Search a product by type (press 3); \n";
        cout<<"4. Search a product by bar code (press 4); \n";
        cout<<"5. Exit (press 0); \n";
        cout<<"Enter your option: ";
        cin>>choice;
        cout<<endl<<endl;
        switch (choice)
        {
            case 1:
            {
                cout<<"Displaying your data..."<<endl;
                cout<<endl;
                displaydata();
                break;
            }
            case 2:
            {
                cout<<"Searching..."<<endl;
                cout<<endl;
                searchbybrand();
                break;
            }
            case 3:
            {
                cout<<"Searching..."<<endl;
                cout<<endl;
                searchbytype();
                break;
            }
            case 4:
            {
                cout<<"Searching..."<<endl;
                cout<<endl;
                searchbycodebar();
                break;
            }
        }
    }while (choice != 0);
    return 0;
}
```



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

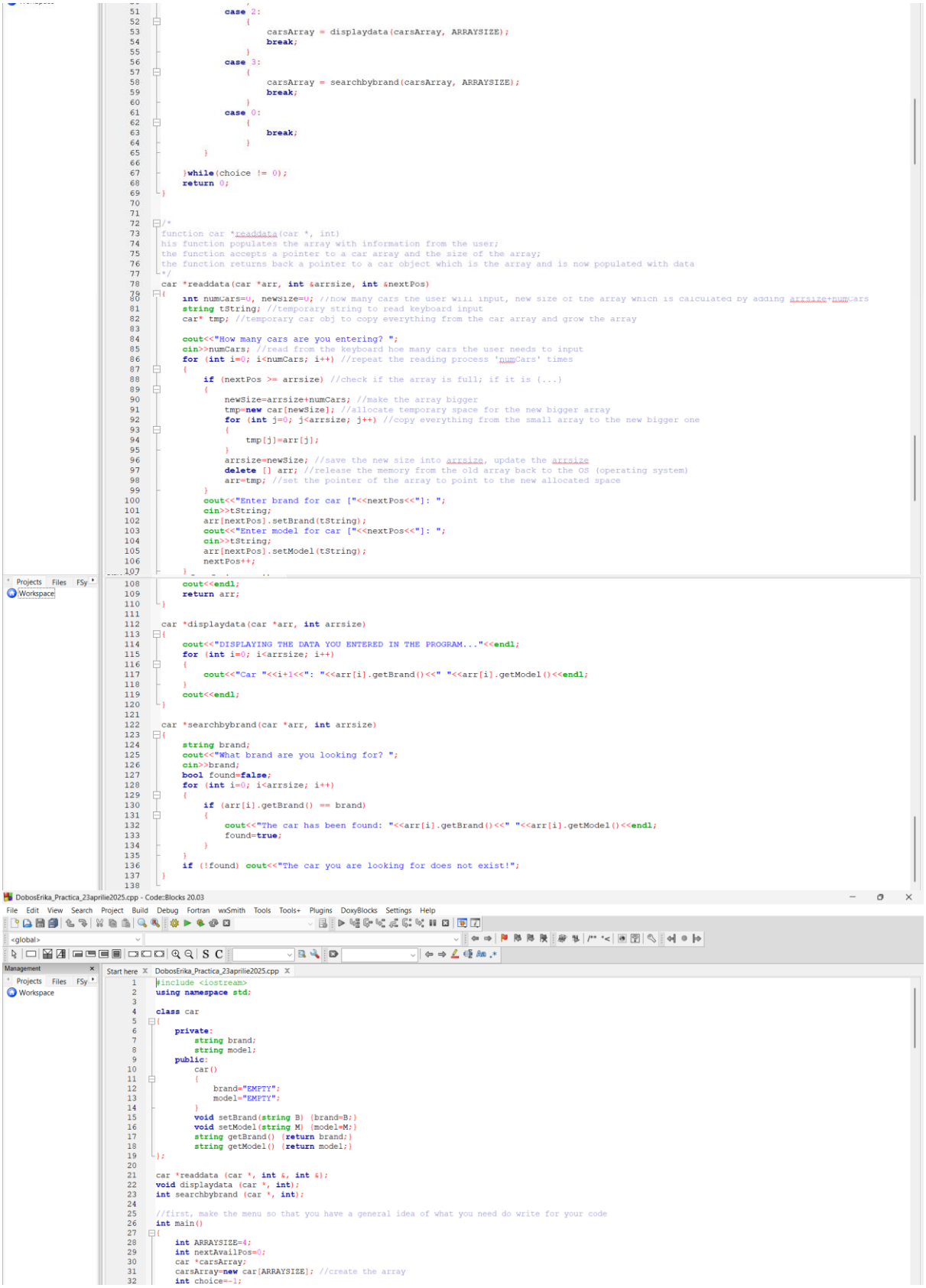
class car
{
    private:
        string brand;
        string model;
    public:
        car()
        {
            brand="EMPTY";
            model="EMPTY";
        }
        void setBrand(string B) {brand=B;}
        void setModel(string M) {model=M;}
        string getBrand() {return brand;}
        string getModel() {return model;}
};

car *readdata(car *, int &, int &);
car *displaydata(car *, int );
car *searchbybrand(car *, int );


//first, make the menu so that you have a general idea of what you need do write fro your code
int main()
{
    int ARRAYSIZE=4;
    int nextAvailPos=0;
    car *carsArray;
    carsArray=new car[ARRAYSIZE]; //create the array
    int choice=-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 the program"<<endl;
        cin>>choice;
        cout<<endl;

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

Dobos Erika



```
            case 2:
                {
                    carsArray = displaydata(carsArray, ARRAYSIZE);
                    break;
                }
            case 3:
                {
                    carsArray = searchbybrand(carsArray, ARRAYSIZE);
                    break;
                }
            case 0:
                {
                    break;
                }
        }

    }while(choice != 0);
    return 0;
}


/*
function car *readdata(car *, int)
his function populates the array with information from the user;
the function accepts a pointer to a car array and the size of the array;
the function returns back a pointer to a car object which is the array and is now populated with data
*/
car *readdata(car *arr, int &arrsize, int &nextPos)
{
    int numCars=0, newSize=0; //how many cars the user will input, new size of the array which is calculated by adding arrsize+numCars
    string tString; //temporary string to read keyboard input
    car* tmp; //temporary car obj to copy everything from the car array and grow the array

    cout<<"How many cars are you entering? ";
    cin>>numCars; //read from the keyboard hoe many cars the user needs to input
    for (int i=0; i<numCars; i++) //repeat the reading process 'numCars' times
    {
        if (nextPos >= arrsize) //check if the array is full; if it is {...}
        {
            newSize=arrsize+numCars; //make the array bigger
            tmp=new car[newSize]; //allocate temporary space for the new bigger array
            for (int j=0; j<arrsize; j++) //copy everything from the small array to the new bigger one
            {
                tmp[j]=arr[j];
            }
            arrsize=newSize; //save the new size into arrsize, update the arrsize
            delete [] arr; //release the memory from the old array back to the OS (operating system)
            arr=tmp; //set the pointer of the array to point to the new allocated space
        }
        cout<<"Enter brand for car ["<<nextPos<<"]: ";
        cin>>tString;
        arr[nextPos].setBrand(tString);
        cout<<"Enter model for car ["<<nextPos<<"]: ";
        cin>>tString;
        arr[nextPos].setModel(tString);
        nextPos++;
    }
    cout<<endl;
    return arr;
}

car *displaydata(car *arr, int arrsize)
{
    cout<<"DISPLAYING THE DATA YOU ENTERED IN THE PROGRAM..."<<endl;
    for (int i=0; i<arrsize; i++)
    {
        cout<<"Car "<<i+1<<": "<<arr[i].getBrand()<<" "<<arr[i].getModel()<<endl;
    }
    cout<<endl;
}

car *searchbybrand(car *arr, int arrsize)
{
    string brand;
    cout<<"What brand are you looking for? ";
    cin>>brand;
    bool found=false;
    for (int i=0; i<arrsize; i++)
    {
        if (arr[i].getBrand() == brand)
        {
            cout<<"The car has been found: "<<arr[i].getBrand()<<" "<<arr[i].getModel()<<endl;
            found=true;
        }
    }
    if (!found) cout<<"The car you are looking for does not exist!";
}
```

DobosErika_Practica_23aprilie2025.cpp - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Start here | DobosErika_Practica_23aprilie2025.cpp

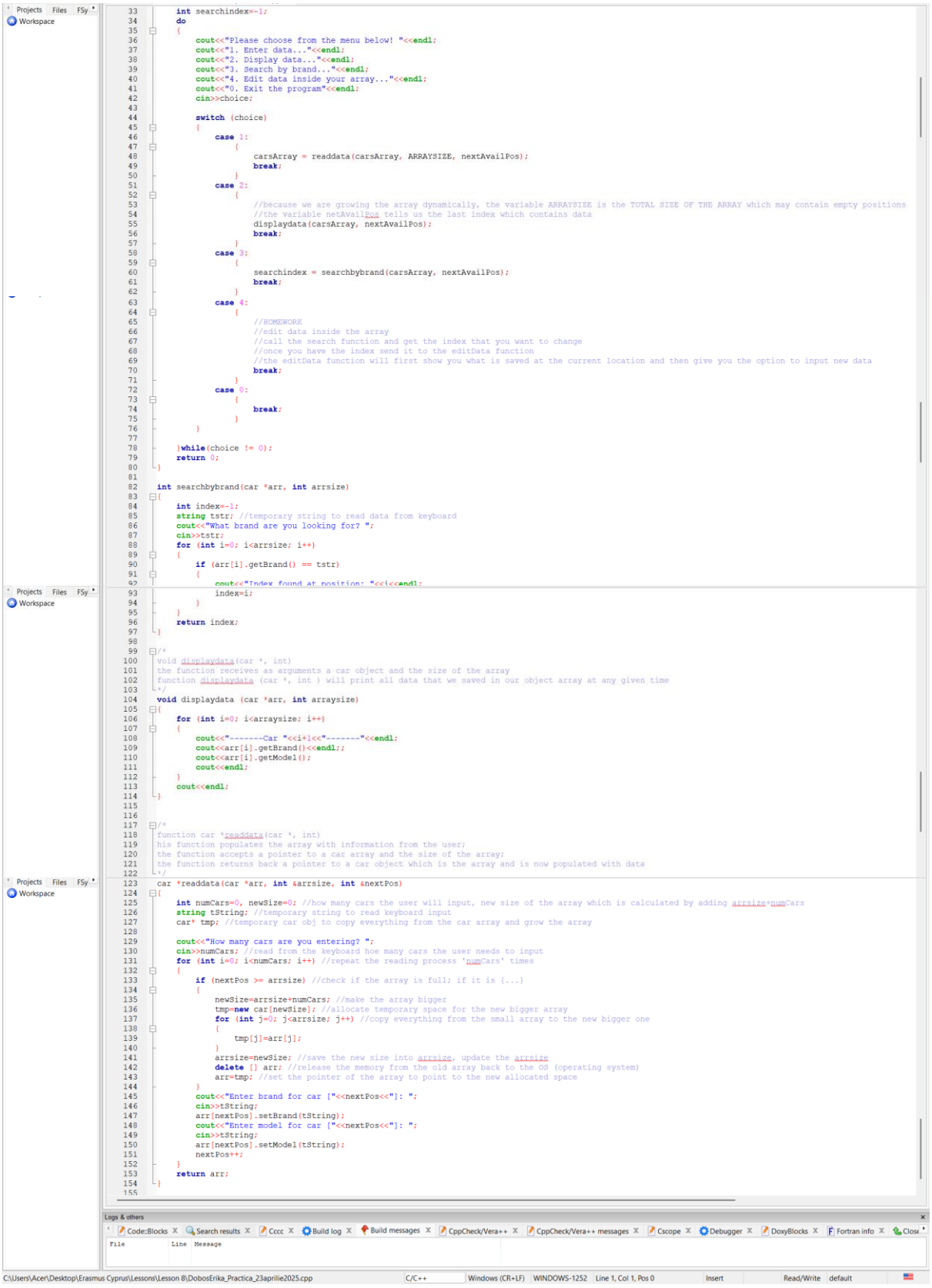
```
#include <iostream>
using namespace std;

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

car *readdata (car *, int &, int &);
void displaydata (car *, int);
int searchbybrand (car *, int);

//first, make the menu so that you have a general idea of what you need do write for your code
int main()
{
    int ARRAYSIZE=4;
    int nextAvailPos=0;
    car *carsArray;
    carsArray=new car[ARRAYSIZE]; //create the array
    int choice=-1;
```

Dobos Erika



```
    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. Edit data inside your array..."<<endl;
        cout<<"0. Exit the program"<<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 netAvailPos tells us the last index which contains data
                    displaydata(carsArray, nextAvailPos);
                    break;
                }
            case 3:
                {
                    searchindex = searchbybrand(carsArray, nextAvailPos);
                    break;
                }
            case 4:
                {
                    //HOMEWORK
                    //edit data inside the array
                    //call the search function and get the index that you want to change
                    //once you have the index send it to the editData function
                    //the editData function will first show you what is saved at the current location and then give you the option to input new data
                    break;
                }
            case 0:
                {
                    break;
                }
        }

    }while(choice != 0);
    return 0;
}

int searchbybrand(car *arr, int arrsize)
{
    int index=-1;
    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<<endl;
            index=i;
        }
    }
    return index;
}

/*
void displaydata(car *, int)
the function receives as arguments a car object and the size of the array
function displaydata (car *, int ) will print all data that we saved in our object array at any given time
*/
void displaydata (car *arr, int arraysize)
{
    for (int i=0; i<arraysize; i++)
    {
        cout<<"-------Car "<<i+1<<"-------"<<endl;
        cout<<arr[i].getBrand()<<endl;;
        cout<<arr[i].getModel();
        cout<<endl;
    }
    cout<<endl;
}


/*
function car *readdata(car *, int)
his function populates the array with information from the user;
the function accepts a pointer to a car array and the size of the array;
the function returns back a pointer to a car object which is the array and is now populated with data
*/
car *readdata(car *arr, int &arrsize, int &nextPos)
{
    int numCars=0, newSize=0; //how many cars the user will input, new size of the array which is calculated by adding arrsize+numCars
    string tString; //temporary string to read keyboard input
    car* tmp; //temporary car obj to copy everything from the car array and grow the array

    cout<<"How many cars are you entering? ";
    cin>>numCars; //read from the keyboard hoe many cars the user needs to input
    for (int i=0; i<numCars; i++) //repeat the reading process 'numCars' times
    {
        if (nextPos >= arrsize) //check if the array is full; if it is {...}
        {
            newSize=arrsize+numCars; //make the array bigger
            tmp=new car[newSize]; //allocate temporary space for the new bigger array
            for (int j=0; j<arrsize; j++) //copy everything from the small array to the new bigger one
            {
                tmp[j]=arr[j];
            }
            arrsize=newSize; //save the new size into arrsize, update the arrsize
            delete [] arr; //release the memory from the old array back to the OS (operating system)
            arr=tmp; //set the pointer of the array to point to the new allocated space
        }
        cout<<"Enter brand for car ["<<nextPos<<"]: ";
        cin>>tString;
        arr[nextPos].setBrand(tString);
        cout<<"Enter model for car ["<<nextPos<<"]: ";
        cin>>tString;
        arr[nextPos].setModel(tString);
        nextPos++;
    }
    return arr;
}
```

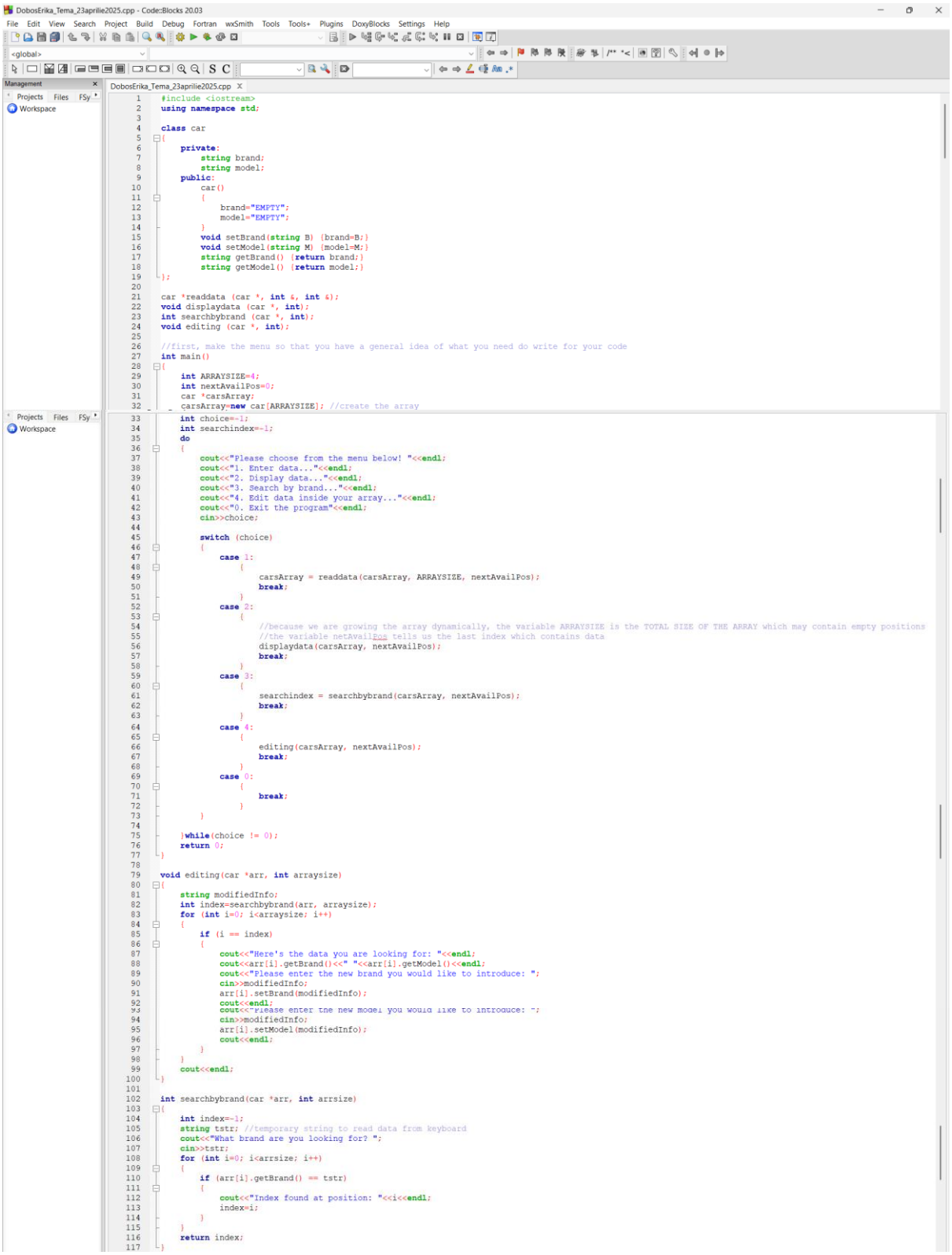
Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppCheck/Vera++ messages | Cscope | Debugger | DoxyBlocks | Fortran info | Close

File | Line | Message

C:\Users\Acer\Desktop\Erasmus Cyprus\Lessons\Lesson 8\DobosErika_Practica_23aprilie2025.cpp | C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 1, Col 1, Pos 0 | Insert | Read/Write | default

Dobos Erika


DobosErika_Tema_23aprilie2025.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Files FSy
Workspace
DobosErika_Tema_23aprilie2025.cpp
#include <iostream>
using namespace std;
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;}
string getModel() {return model;}
};
car *readdata (car *, int &, int &);
void displaydata (car *, int);
int searchbybrand (car *, int);
void editing (car *, int);
//first, make the menu so that you have a general idea of what you need do write for your code
int main()
{
int ARRAYSIZE=4;
int nextAvailPos=0;
car *carsArray;
carsArray=new car[ARRAYSIZE]; //create the array
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. Edit data inside your array..."<<endl;
cout<<"0. Exit the program"<<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;
}
case 3:
{
searchindex = searchbybrand(carsArray, nextAvailPos);
break;
}
case 4:
{
editing(carsArray, nextAvailPos);
break;
}
case 0:
{
break;
}
}
}while(choice != 0);
return 0;
}
void editing(car *arr, int arraysize)
{
string modifiedInfo;
int index=searchbybrand(arr, arraysize);
for (int i=0; i<arraysize; i++)
{
if (i == index)
{
cout<<"Here's the data you are looking for: "<<endl;
cout<<arr[i].getBrand()<<" "<<arr[i].getModel()<<endl;
cout<<"Please enter the new brand you would like to introduce: ";
cin>>modifiedInfo;
arr[i].setBrand(modifiedInfo);
cout<<endl;
cout<<"Please enter the new model you would like to introduce: ";
cin>>modifiedInfo;
arr[i].setModel(modifiedInfo);
cout<<endl;
}
}
cout<<endl;
}
int searchbybrand(car *arr, int arrsize)
{
int index=-1;
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<<endl;
index=i;
}
}
return index;
}

Dobos Erika



```
/*
void displaydata(car *, int)
the function receives as arguments a car object and the size of the array
function displaydata (car *, int ) will print all data that we saved in our object array at any given time
*/
void displaydata (car *arr, int arraysize)
{
    for (int i=0; i<arraysize; i++)
    {
        cout<<"-------Car "<<i+1<<"-------"<<endl;
        cout<<arr[i].getBrand()<<endl;;
        cout<<arr[i].getModel();
        cout<<endl;
    }
    cout<<endl;
}


/*
function car *readdata(car *, int)
his function populates the array with information from the user;
the function accepts a pointer to a car array and the size of the array;
the function returns back a pointer to a car object which is the array and is now populated with data
*/
car *readdata(car *arr, int &arrsize, int &nextPos)
{
    int numCars=0, newSize=0; //how many cars the user will input, new size of the array which is calculated by adding arrsize+numCars
    string tString; //temporary string to read keyboard input
    car* tmp; //temporary car obj to copy everything from the car array and grow the array

    cout<<"How many cars are you entering? ";
    cin>>numCars; //read from the keyboard hoe many cars the user needs to input
    for (int i=0; i<numCars; i++) //repeat the reading process 'numCars' times
    {
        if (nextPos >= arrsize) //check if the array is full; if it is {...}
        {
            newSize=arrsize+numCars; //make the array bigger
            tmp=new car[newSize]; //allocate temporary space for the new bigger array
            for (int j=0; j<arrsize; j++) //copy everything from the small array to the new bigger one
            {
                tmp[j]=arr[j];
            }
            arrsize=newSize; //save the new size into arrsize, update the arrsize
            delete [] arr; //release the memory from the old array back to the OS (operating system)
            arr=tmp; //set the pointer of the array to point to the new allocated space
        }
        cout<<"Enter brand for car ["<<nextPos<<"]: ";
        cin>>tString;
        arr[nextPos].setBrand(tString);
        cout<<"Enter model for car ["<<nextPos<<"]: ";
        cin>>tString;
        arr[nextPos].setModel(tString);
        nextPos++;
    }
    cout<<endl;
    return arr;
}

```