

Utilizarea compilatorului Microsoft Visual Studio

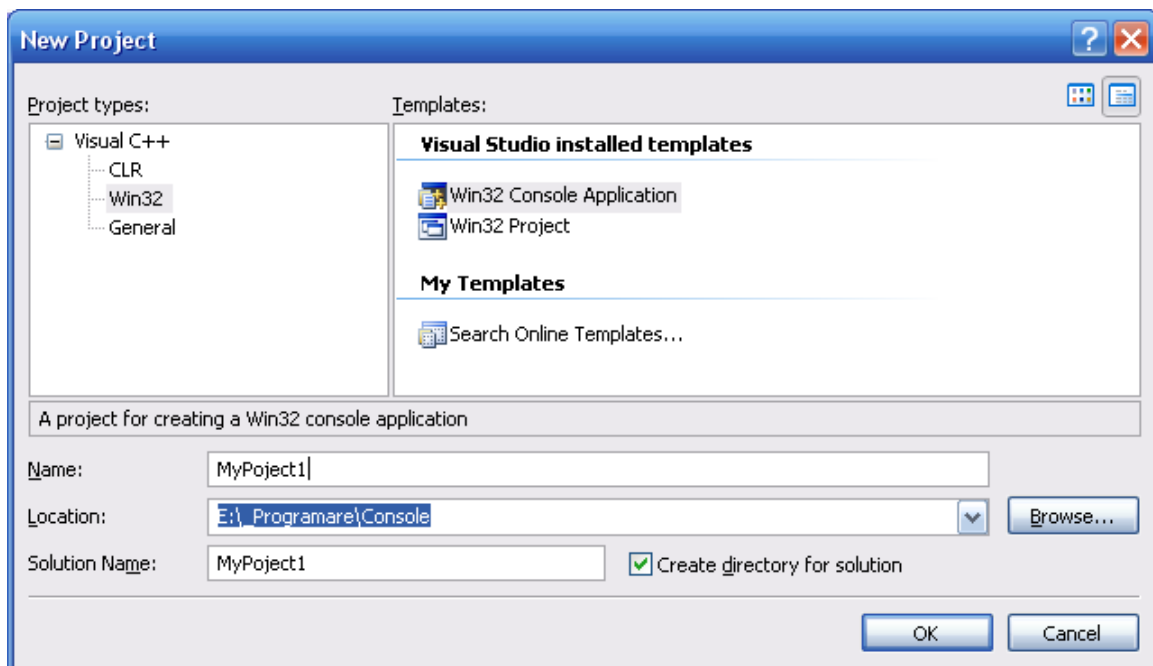
Firma Microsoft pune la dispoziția celor interesați varianta *free* a mediului său de dezvoltare *Visual Studio*, variantă numită *Visual Studio Community*, care poate fi descărcată de la adresa: <https://visualstudio.microsoft.com/downloads/>

Deoarece această variantă este prea amplă pentru începători și este mare consumatoare de resurse, recomand celor interesați să își instaleze una dintre edițiile mai vechi, așa numitele ediții expres, cum ar fi *Visual Studio Express for Windows Desktop 2012 - 2017*, sau chiar *Visual Studio 2008 - 2010*, în funcție de varianta sistemului de operare Windows folosit și de resursele calculatorului personal, următorul tutorial fiind valabil pentru toate aceste ediții.

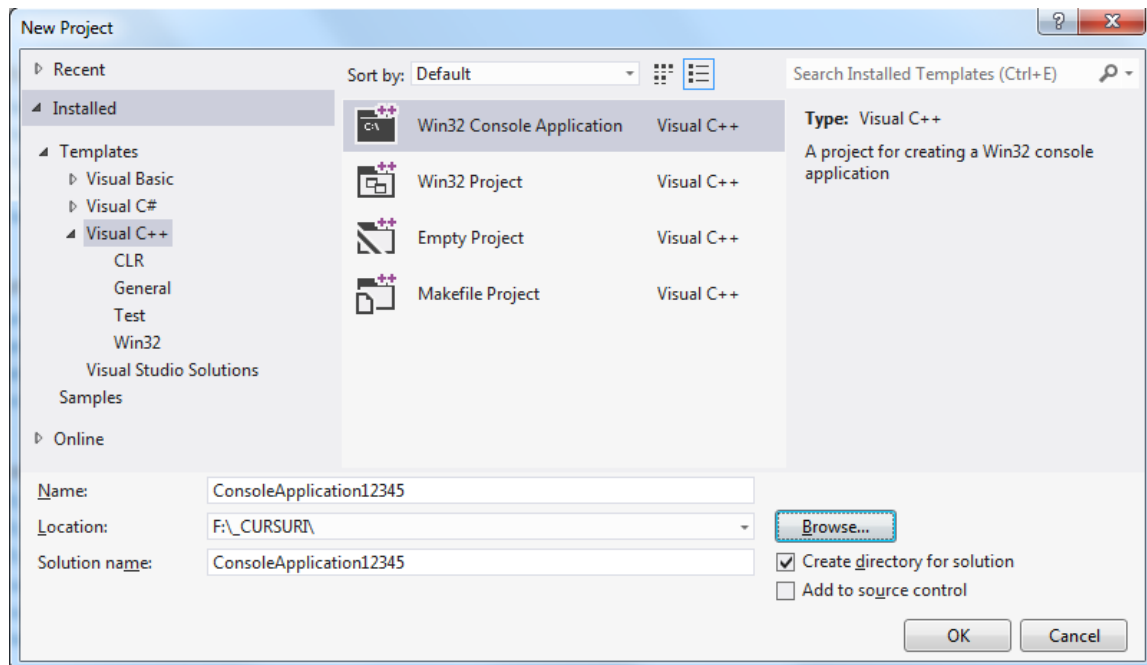
Rularea unui program C/C++ în mod consolă cu Microsoft Visual Studio se execută în patru etape:

1) **Inițierea unui proiect** are loc în doi pași: alegerea proiectului și stabilirea opțiunilor.

Alegerea proiectului se execută din meniul **File, New | Project**. În **Microsoft Visual C++ 2008/2010** apare o fereastră de forma



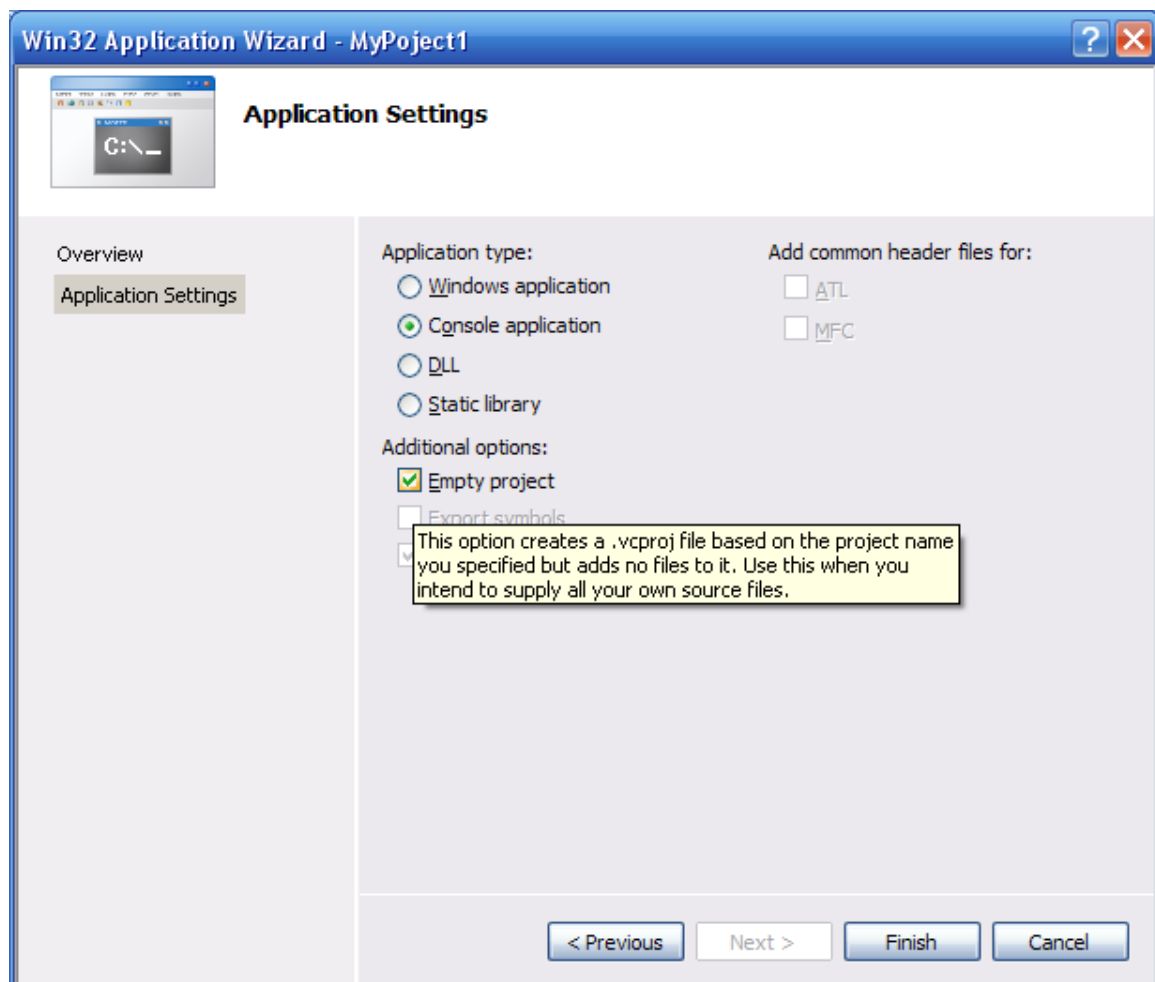
iar în **Microsoft Visual Studio 2012/2015** apare

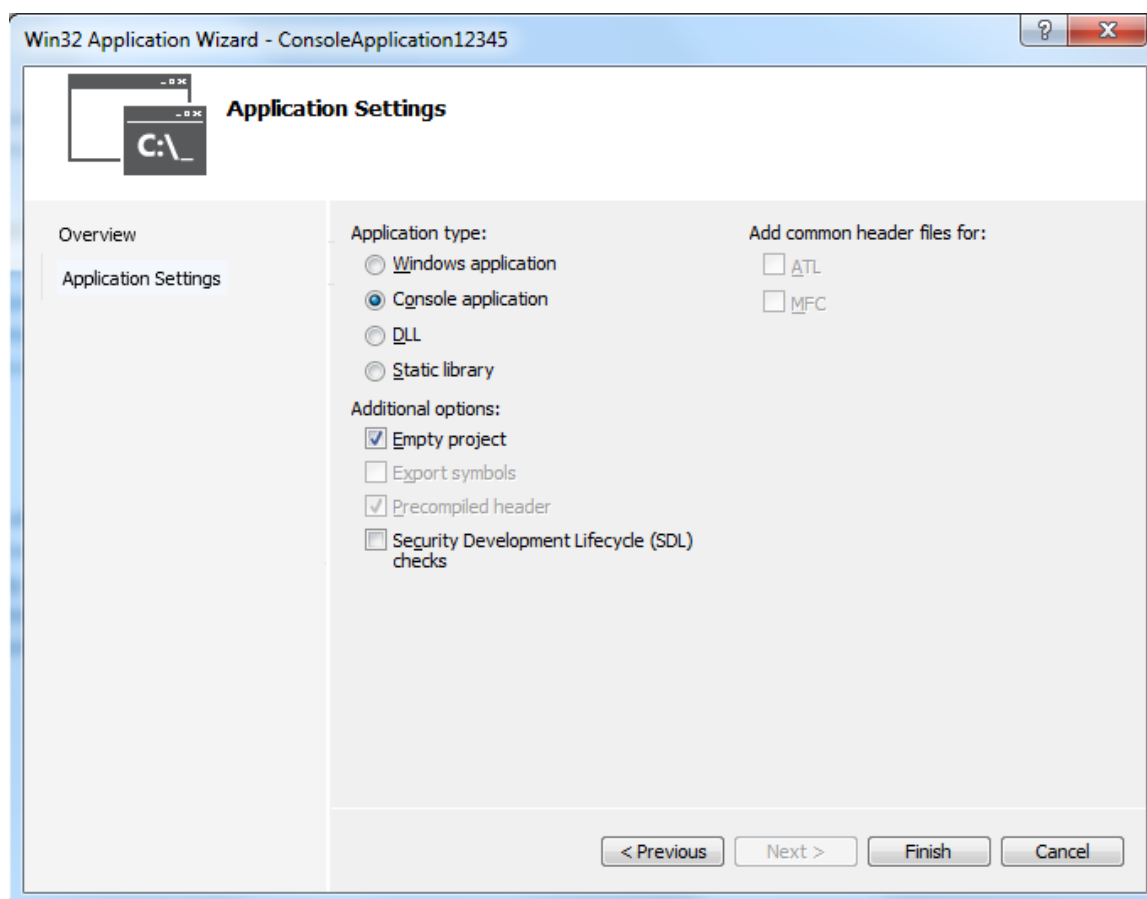


În ambele cazuri alegem un proiect de tipul **Visual C++ Win32** având ca șablon (*template*) **Win32 Console Application**. Dăm un nume proiectului și-i stabilim o locație pe disc.

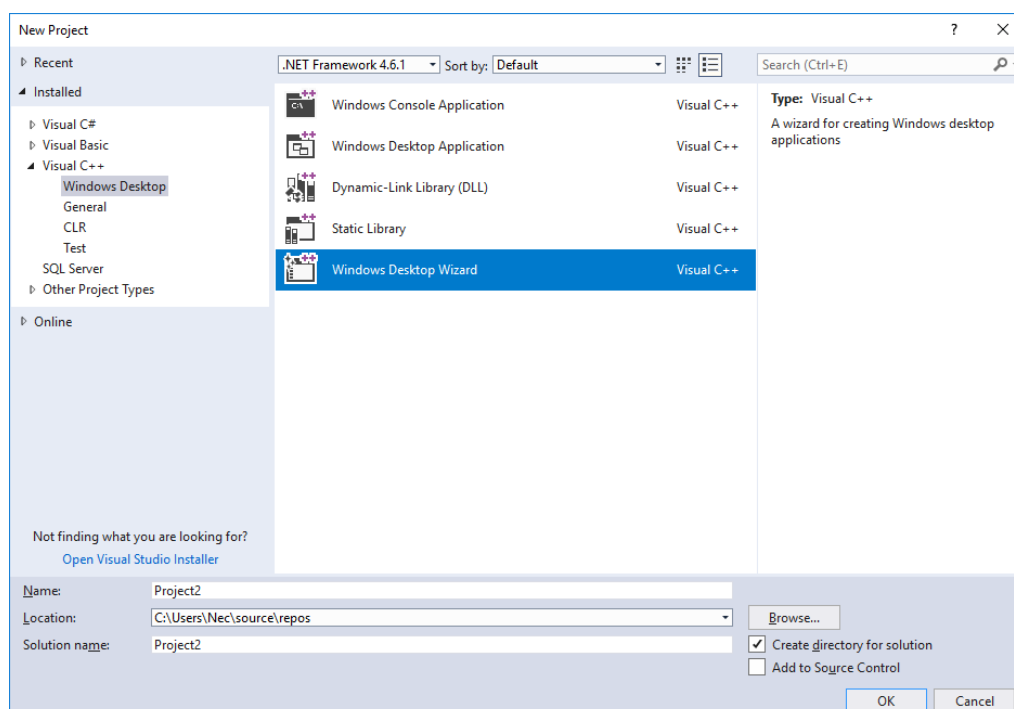
MS Visual Studio folosește noțiunea de *solution* cu sensul de spațiu de lucru (*workspace*), o soluție poate conține mai multe proiecte iar fiecare proiect poate avea mai multe fișiere sursă. Noi vom lucra în stilul 1, 1, 1 (o soluție, un proiect, un singur fișier sursă).

Apăsăm butonul **Ok** și ajungem la pasul 2, fixarea opțiunilor: în următoarea fereastră de dialog alegem **Next** și apoi, atenție, bifăm căsuța „**Empty project**” înainte de „**Finish**”:

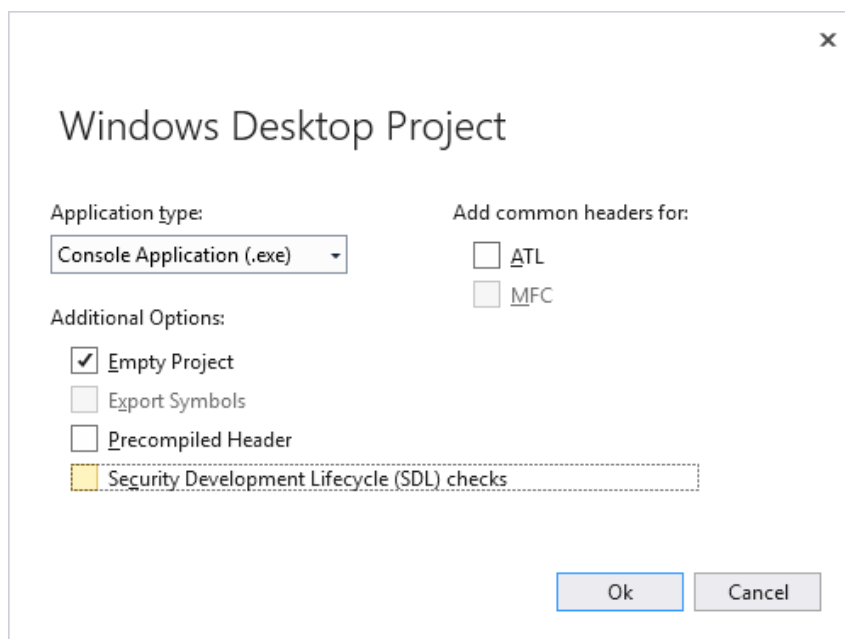




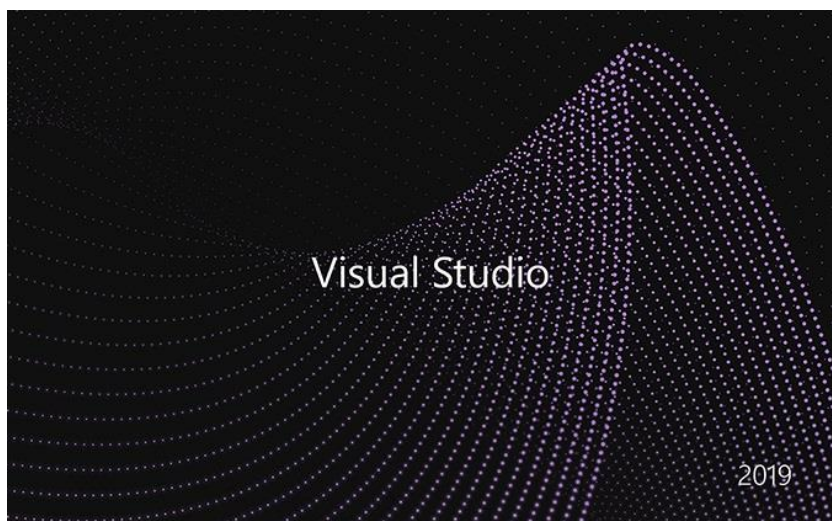
În cazul compilatorului **Microsoft Visual Studio 2017**, la pasul 1, în fereastra New Project, se alege Visual C++/ Windows Desktop în panoul din stânga și Windows Desktop Wizard în cel din centru:



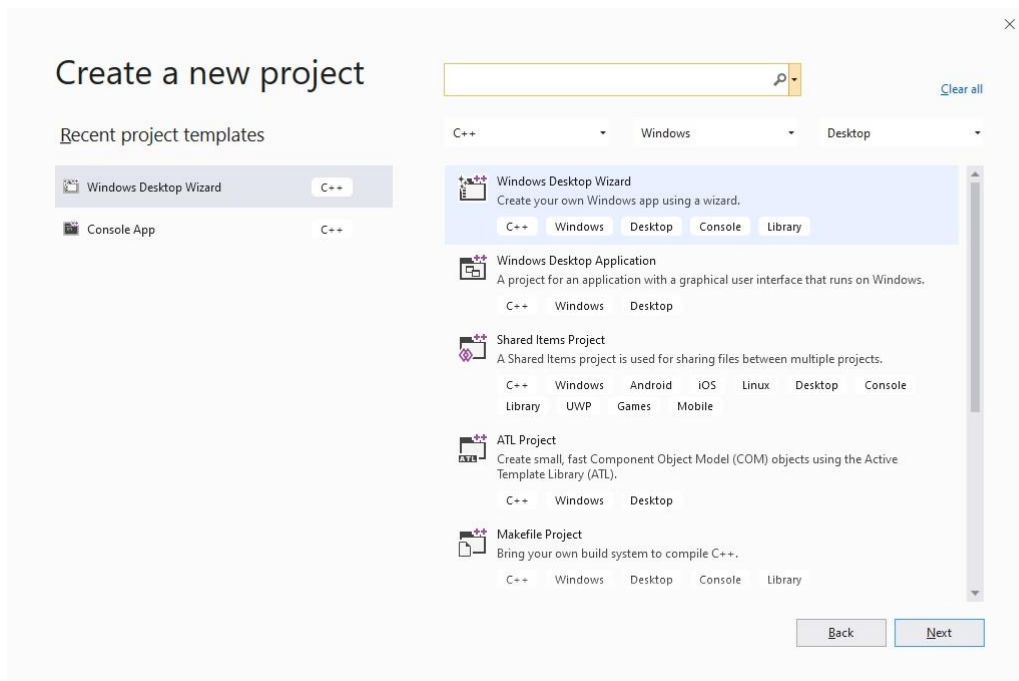
și apoi, la pasul 2, în fereastra Windows Desktop Project, la Application type alegem Console Application (.exe) iar la Additional Options bifăm căsuța „Empty project” și le debifăm toate celelalte.



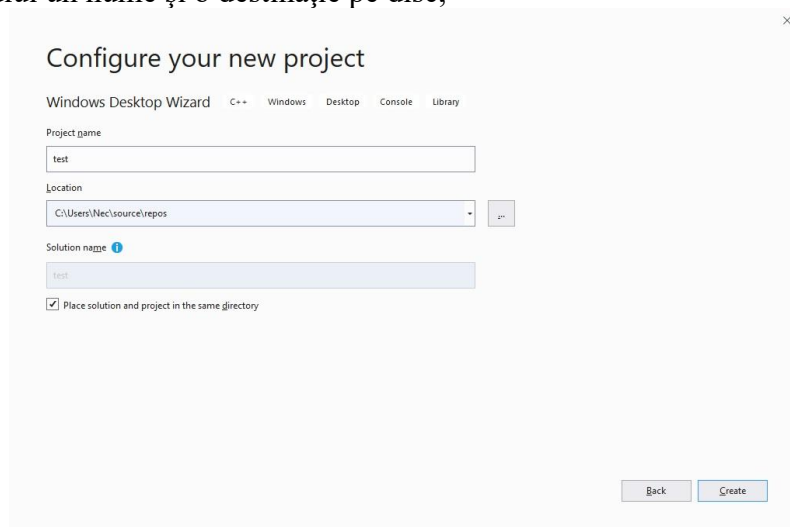
Pentru versiunea, **Microsoft Visual Studio 2019**,



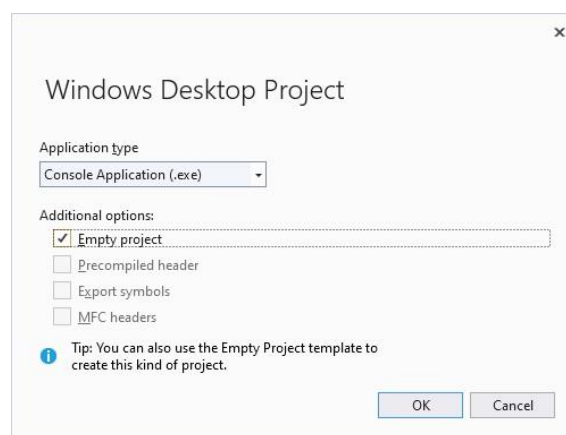
procedăm astfel: din pagina de start alegem butonul **Create a new project**, iar din pagina următoare ne alegem șablonul (*template*) Windows Desktop Wizard, fixând cele trei liste derulante în pozițiile C++, Windows și Desktop, ca în imaginea următoare:



Apoi dăm proiectului un nume și o destinație pe disc,

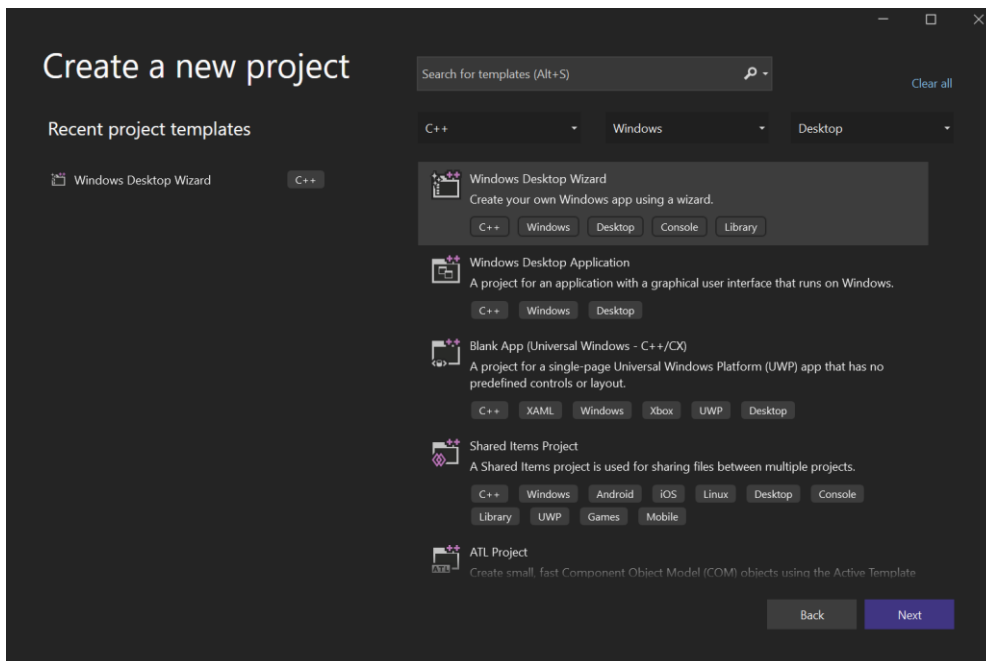


și, în final, bifăm opțiunea **Empty project**:

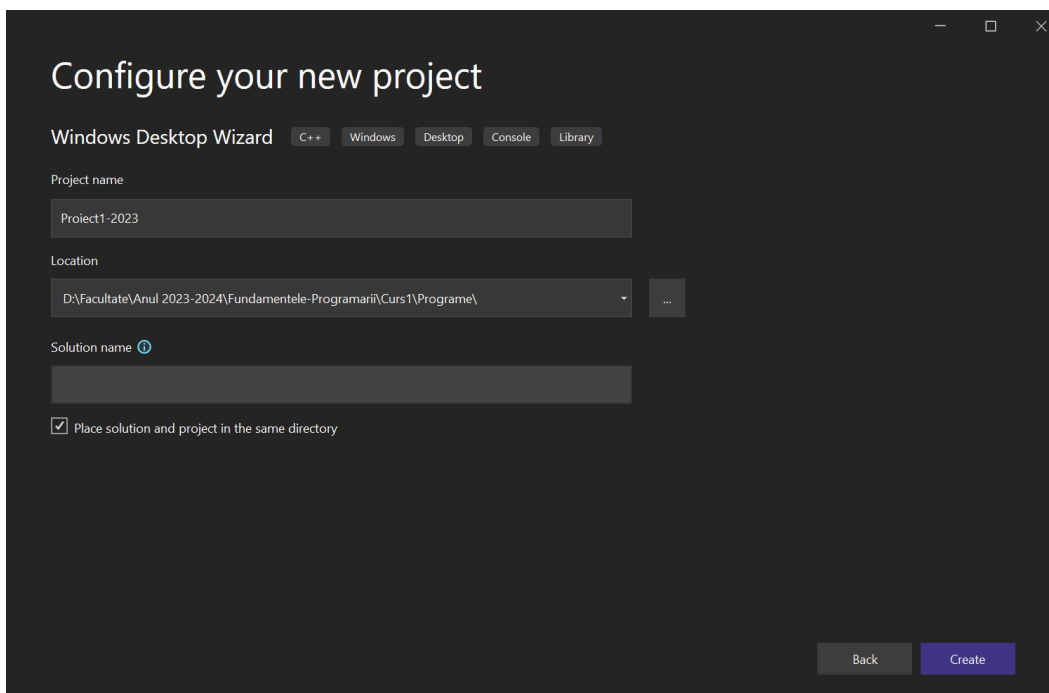


Observație: în ultima imagine apare un sfat (*You can also use ...*), încercați să-l urmați!

Pentru ultima versiune **Microsoft Visual Studio 2022**, procedăm astfel: din pagina de start alegem butonul **Create a new project**, apoi ne alegem șablonul (*template*) Windows Desktop Wizard (alegem limbajul C++, platforma Windows, tipul proiectului Desktop), ca și în imaginea de mai jos

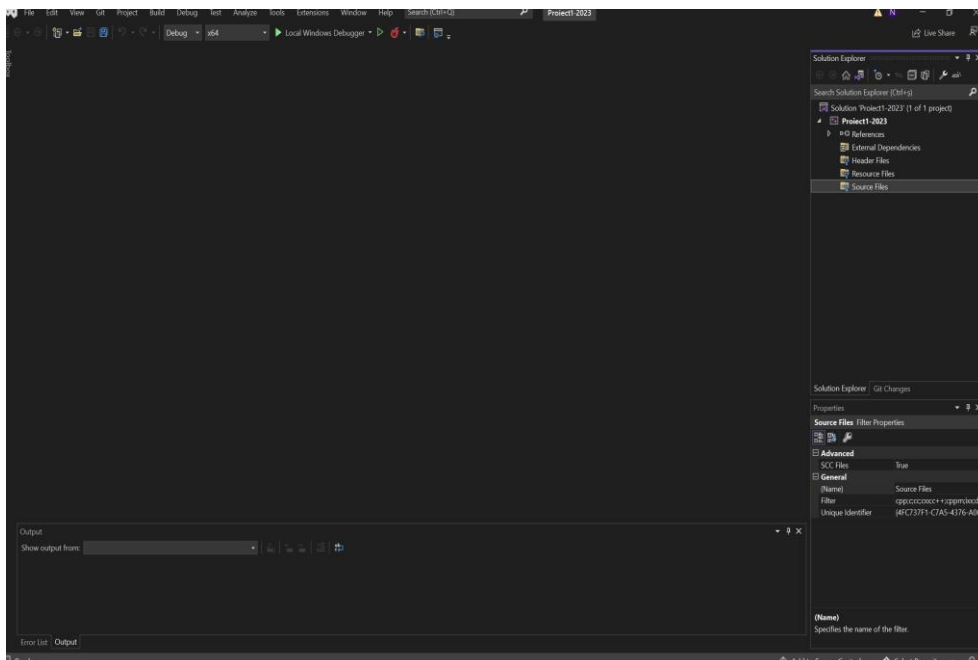
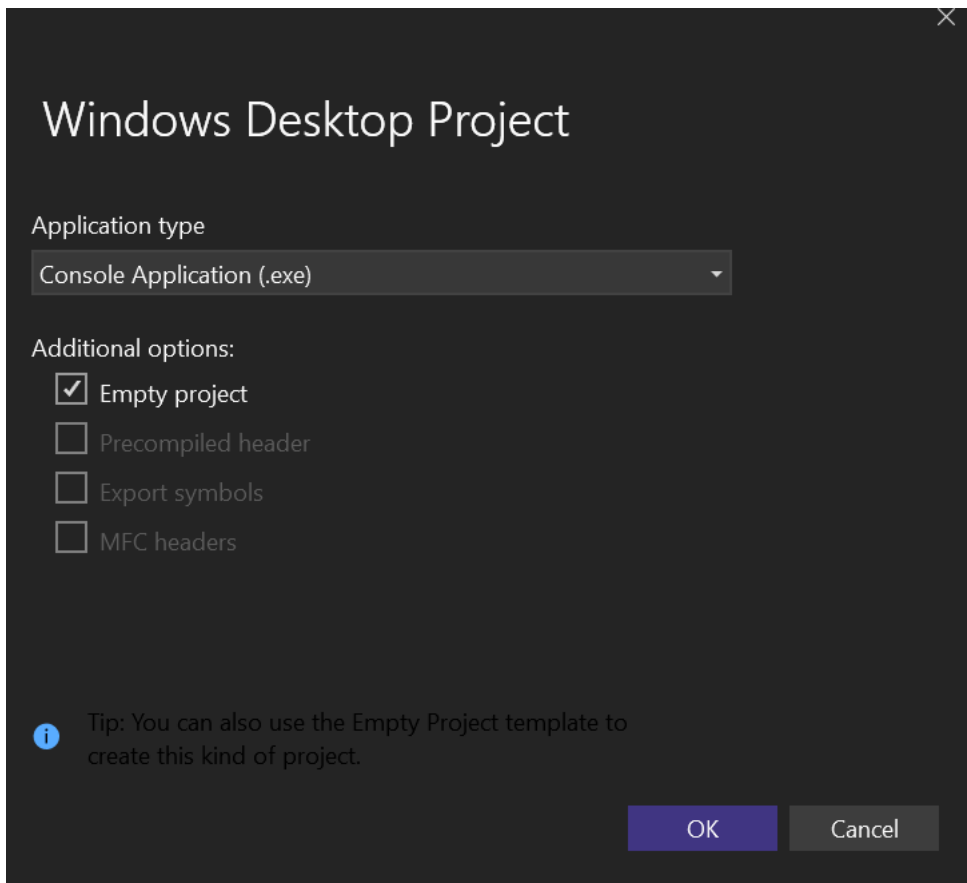


Apăsăm *Next*, apoi îi dăm proiectului un nume și o destinație pe disc și bifăm să se amplaseze sursa și proiectul în același director, după care apăsăm *Create*



În noua fereastră care se deschide, selectăm *Console Application*, bifăm *Empty project*, apoi apăsăm

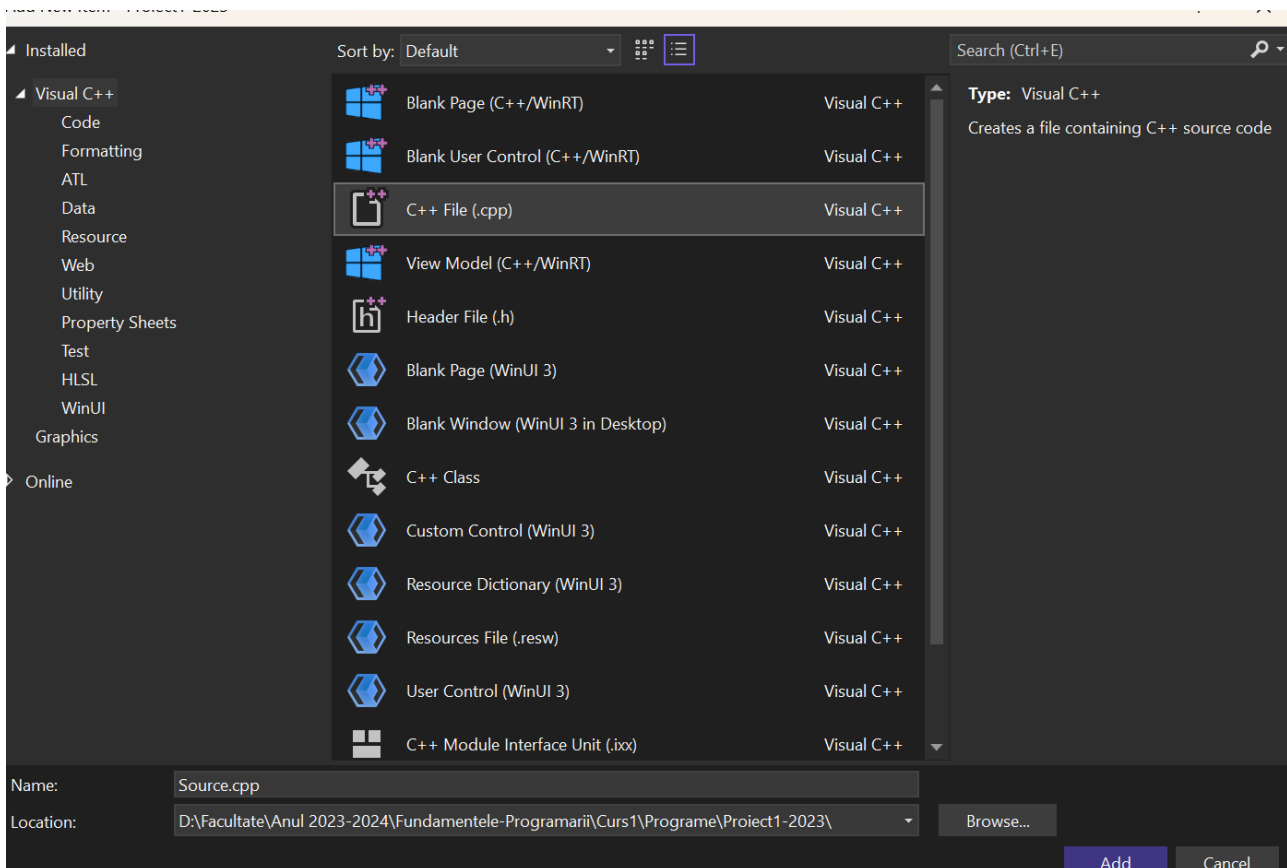
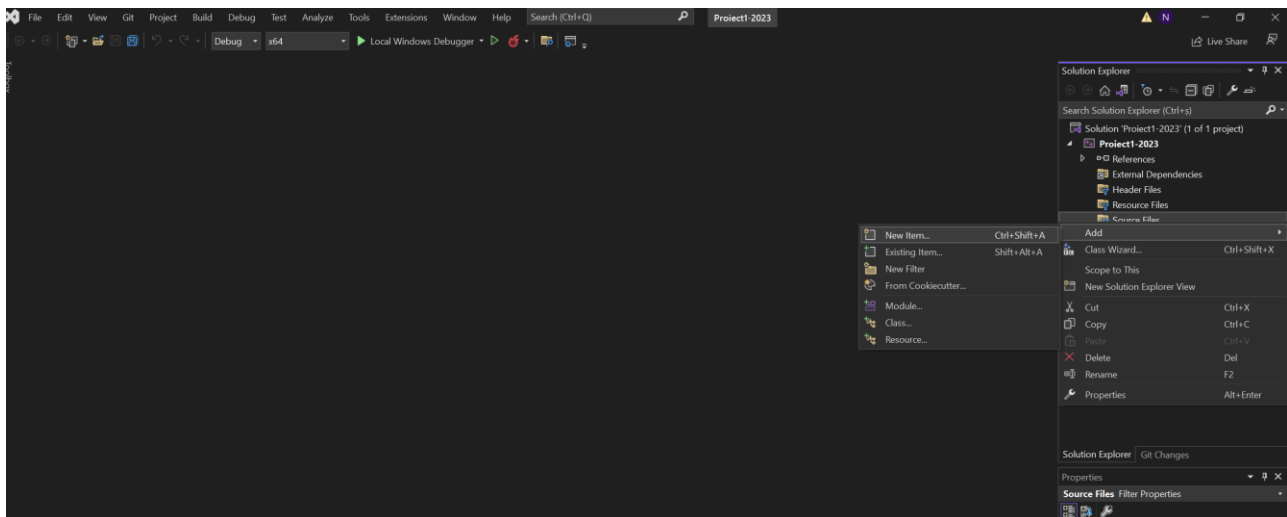
pe *OK*.



2) Adăugarea fișierului sursă.

În fereastra „Solution Explorer” a mediului de dezvoltare facem click dreapta pe „Source Files” (sau deschidem meniul **Project**) și urmăm calea **Add | New Item** și alegem **C++File (.cpp)**. Dăm

un nume fișierului sursă și apasăm butonul „Add”.



3) Editarea textului sursă. Tastăm textul sursă al programului în fereastra editorului de texte. Ca noutate față de compilatoarele mai vechi, biblioteca `<iostream.h>` a fost scoasă din uz, în locul ei vom utiliza versiunea `<iostream>` (fără extensia `.h`) cu directiva de precompilare

```
#include<iostream>
```

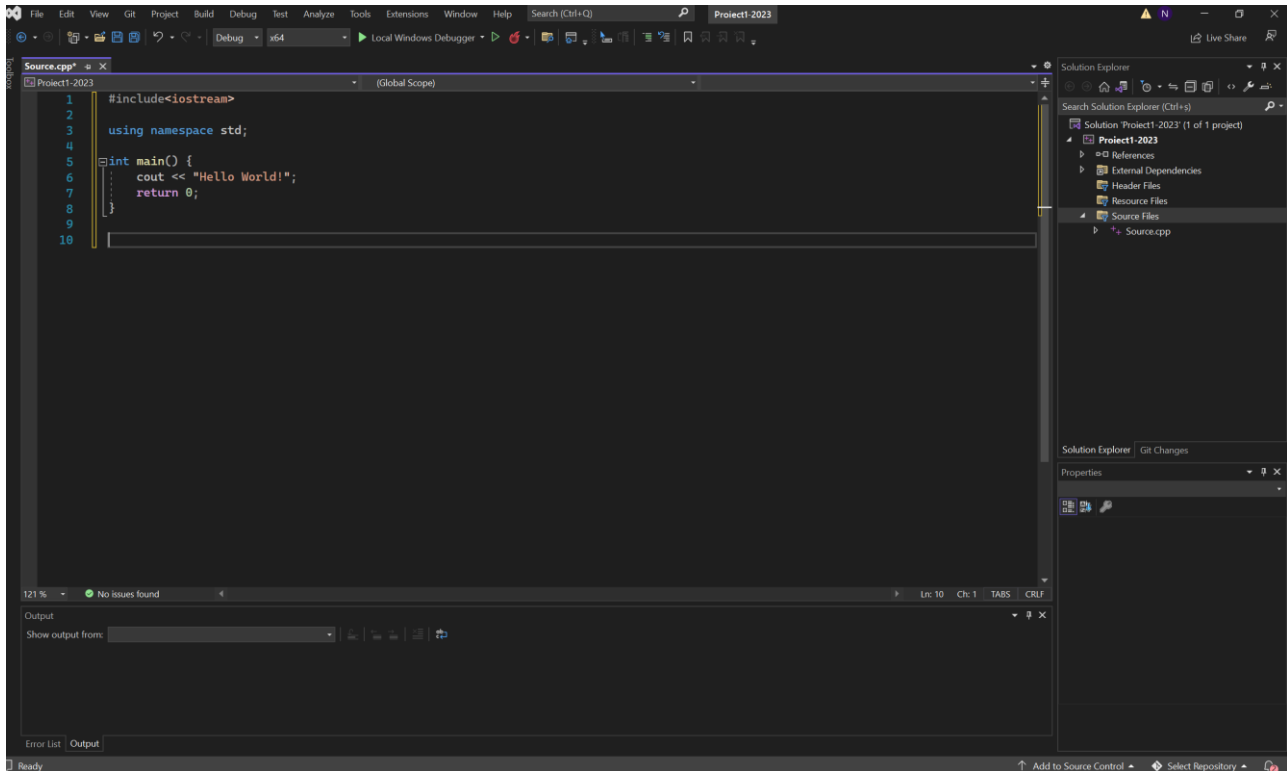
Celelalte biblioteci uzuale au rămas neschimbate.

Imediat după directivele de precompilare vom tasta instrucțiunea:

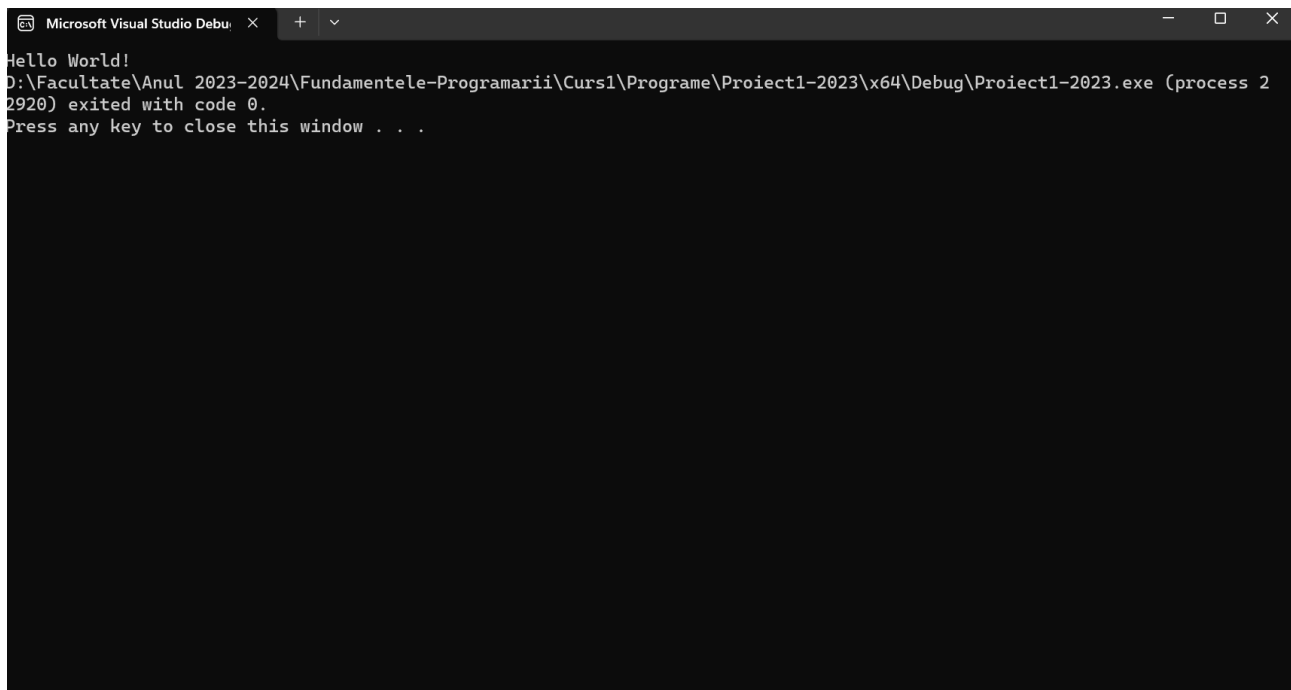
```
using namespace std;
```

care spune compilatorului că utilizăm „spațiul de nume” predefinit `std`, fără această instrucțiune anumiți identificatori ar trebui clarificați cu ajutorul operatorului de rezoluție `::`, de exemplu în loc de `cout` ar trebui scris `std::cout`.

Pentru mai multe informații privind spațiile de nume se poate utiliza MSDN Library (vezi <https://docs.microsoft.com/en-us/cpp/cpp/namespaces-cpp>) sau, de exemplu, următorul tutorial online: <http://www.cplusplus.com/doc/tutorial/namespaces.html> .



4) Compilarea și rularea. Din meniul **Build | Compile (Ctrl F7)** compilăm programul și, după corectarea inerentelor erori de editare, rularea programului se lansează din **Debug | Start without debugging (Ctrl F5)**.



Încercați să rulați următorul exemplu:

```
#include<iostream>
using namespace std;
int main(){
    int i=0;
    while(++i<10000)
        cout<<"Gata, de acum stiu tot! ";
    return 0;
}
```

Mult succes!