

# Getting Started with Microsoft Visual Studio 2008

Acest document este similar cu cel de la POO și oferă informații minimale relative la modul de lucru în VS2008 (crearea, execuția și depănarea unui proiect). Cunoștințele dobândite la laboratoarele de POO vă vor ajuta substanțial pe parcurs.

## 1. Crearea unui proiect în Microsoft Visual Studio 2008

Visual Studio 2008 este mediul de dezvoltare integrat (*IDE – Integrated Development Environment*) pe care îl vom folosi pentru scrierea programelor C++. El permite dezvoltarea de aplicații în mai multe limbaje de programare, de diverse tipuri. În cadrul laboratorului vom scrie programe C++ de tip consolă. Pentru a scrie un program în VS2008, la început trebuie să creăm un proiect.

Pentru a crea un proiect:

1. Pornim VS2008
2. În meniul principal, mergem la *File* → *New* → *Project...* ca în figură:

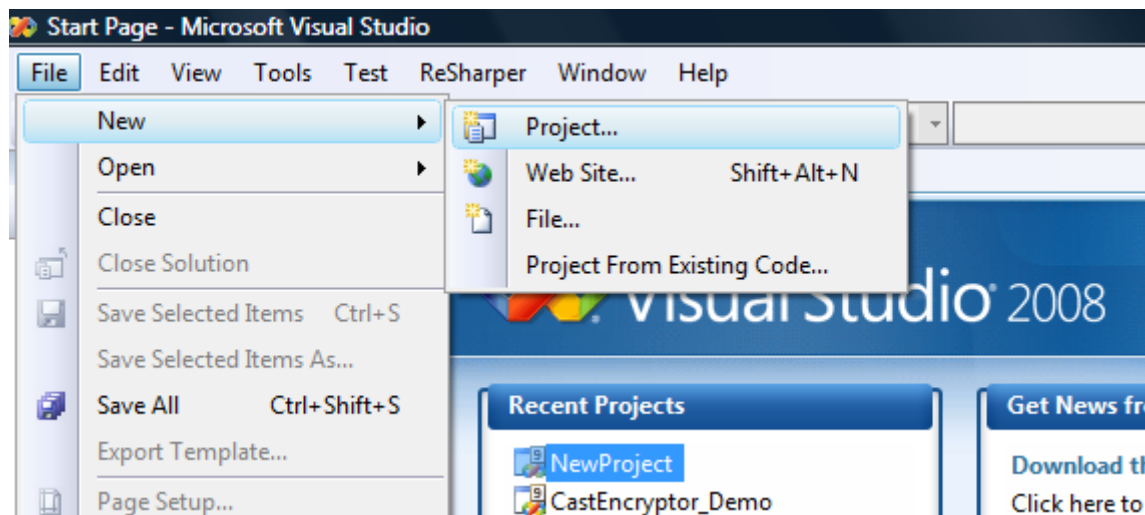


Fig. 1 Creare proiect nou

3. În fereastra New Project (fig. 2), în stânga există căsuța Project types. Alegem *Other languages* → *Visual C++*
4. În partea dreaptă este căsuța Templates. Alegem *Win32 Console Application*.
5. În partea de jos sunt căsuțele Name și Location. Alegem un nume pentru proiect. La Location selectăm un director creat de noi în C:/temp sau D:/temp după caz. (fig. 2)
6. Apăsăm OK.

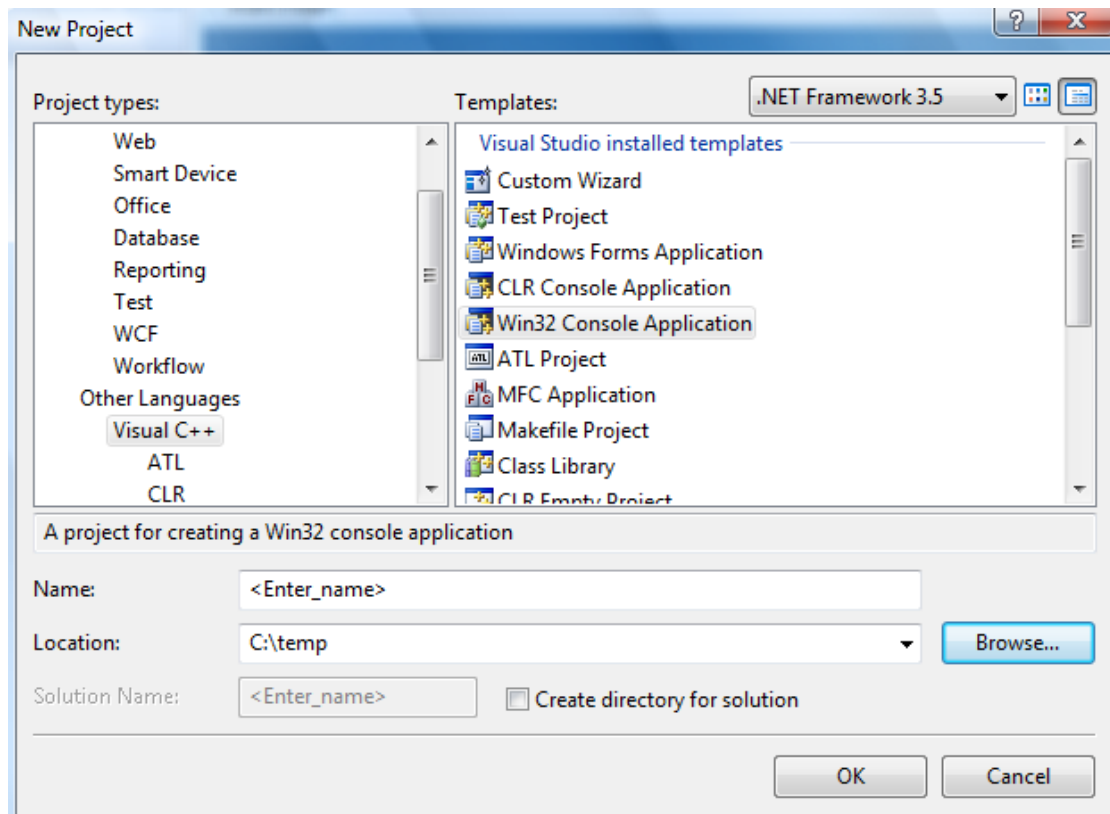


Fig. 2 Fereastra New Project

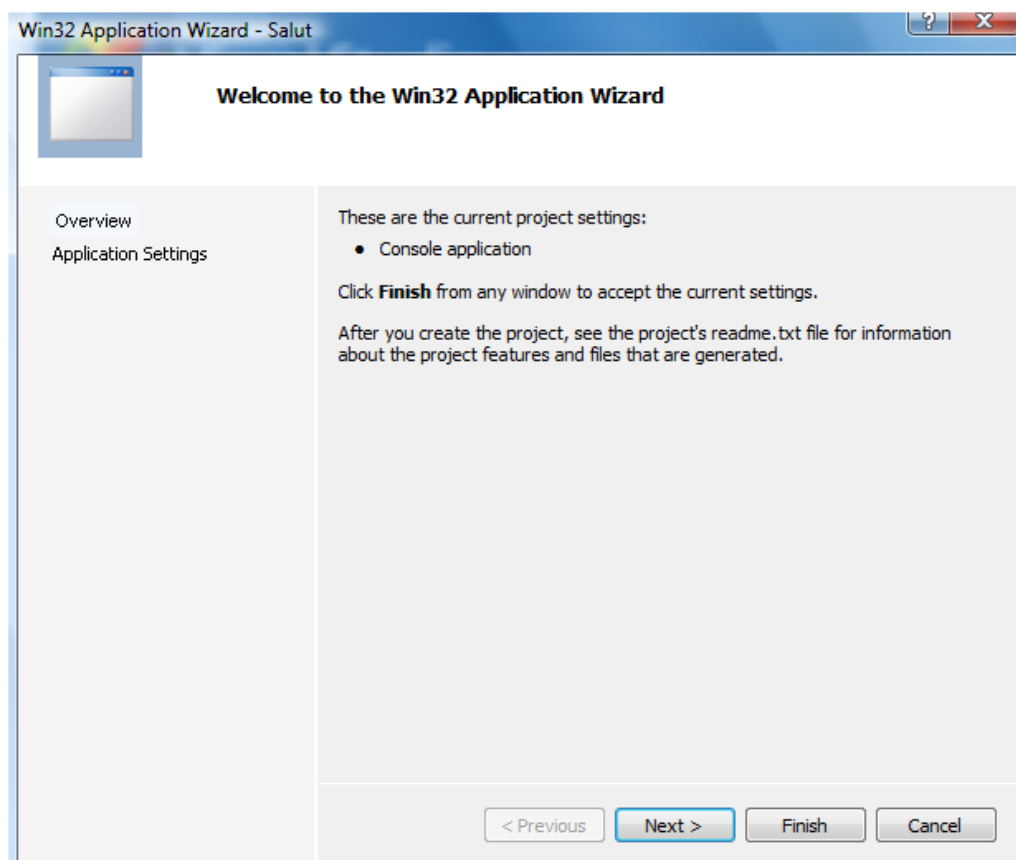
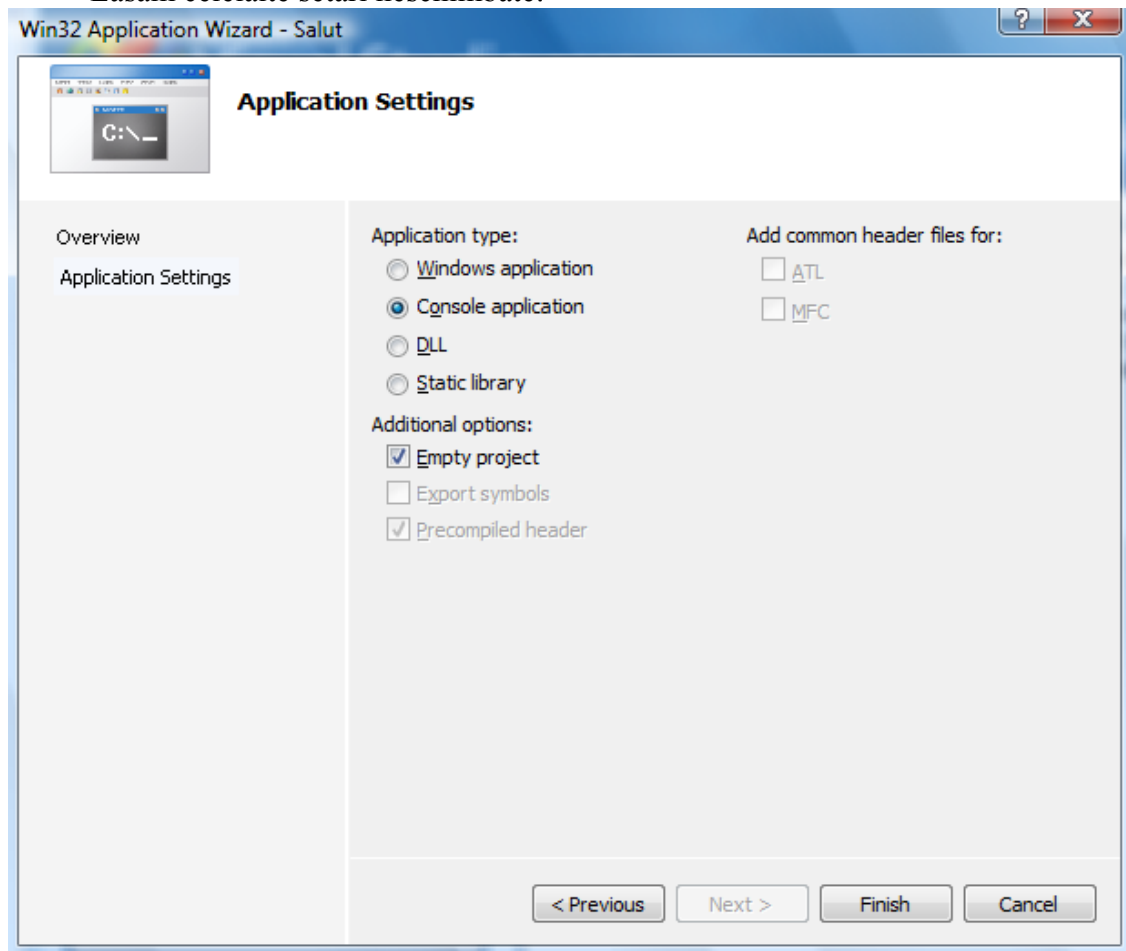


Fig. 3 Fereastra Welcome...

7. Apare fereastra “Welcome to the Win32 Application Wizard” (fig. 3). Apăsăm *Next*, *NU* Finish!
8. În fereastra apărută (fig. 4), rubrica *Additional options*, bifăm “Empty project”. Lăsăm celelalte setări neschimbate:



**Fig. 4 Fereastra Application Settings**

9. Apăsăm *Finish*.  
Proiectul este creat și deschis în VS. Ar trebui să vedem următoarele ferestre:
  - Solution explorer – în partea stânga. De aici se pot crea sau deschide fișierele proiectului. Inițial proiectul nu conține nici un fișier.
  - Start Page – în restul ecranului. Nu ne este de mare folos, putem să o închidem.Mediul de dezvoltare va arăta în felul următor:

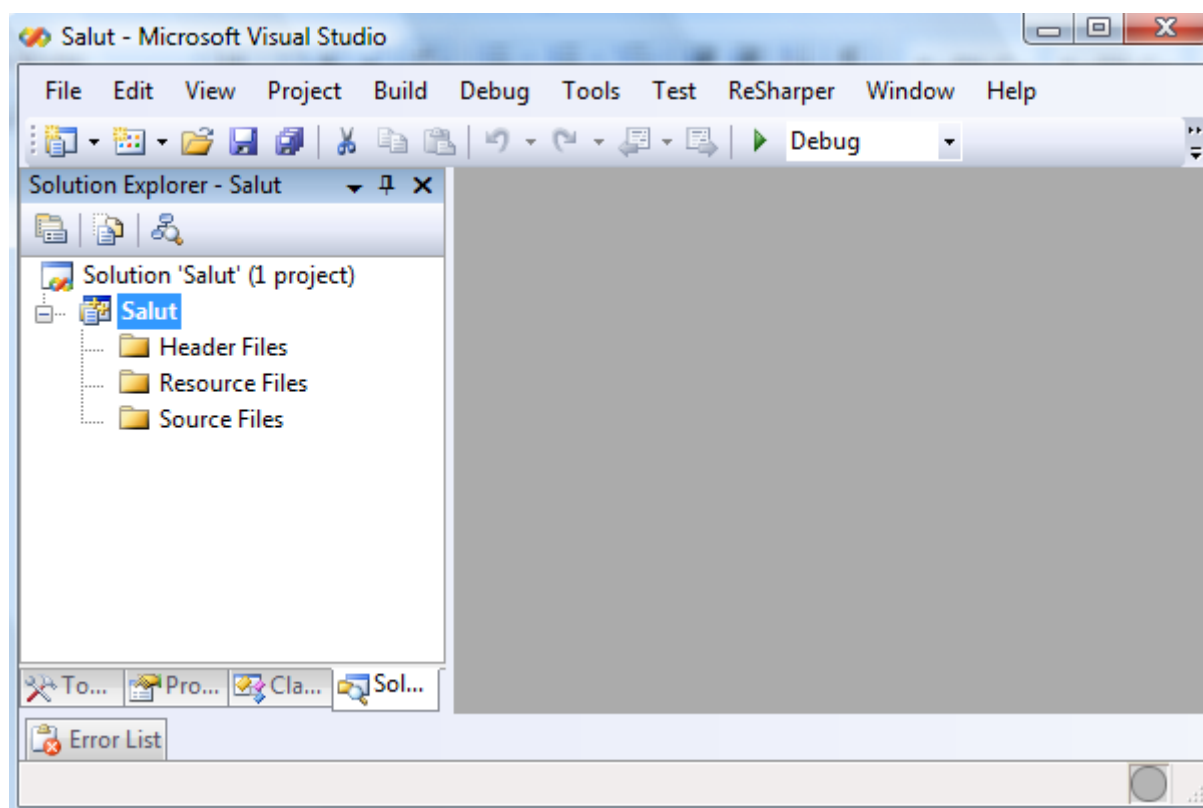


Fig. 5 Proiect nou creat

## 2. Adăugarea unui fișier la proiect

Ca să putem scrie cel mai simplu program în VS2008, trebuie să adăugăm un fișier sursă. Pentru aceasta, urmați următorii pași:

1. În *Solution Explorer*, click dreapta peste grupul *Source Files* → *Add* → *New Item...*

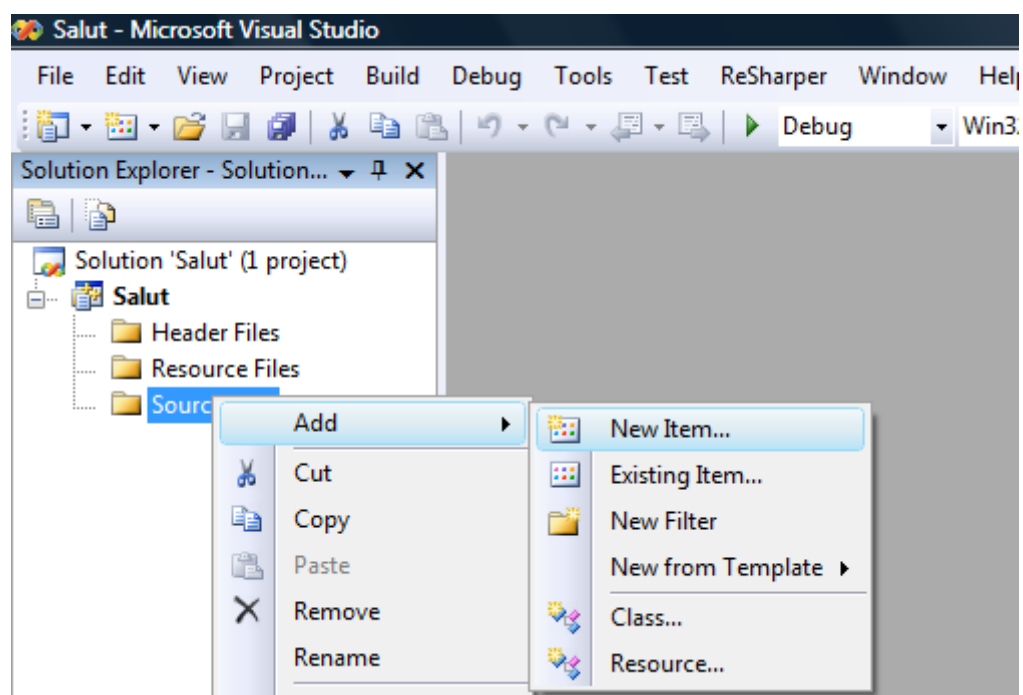
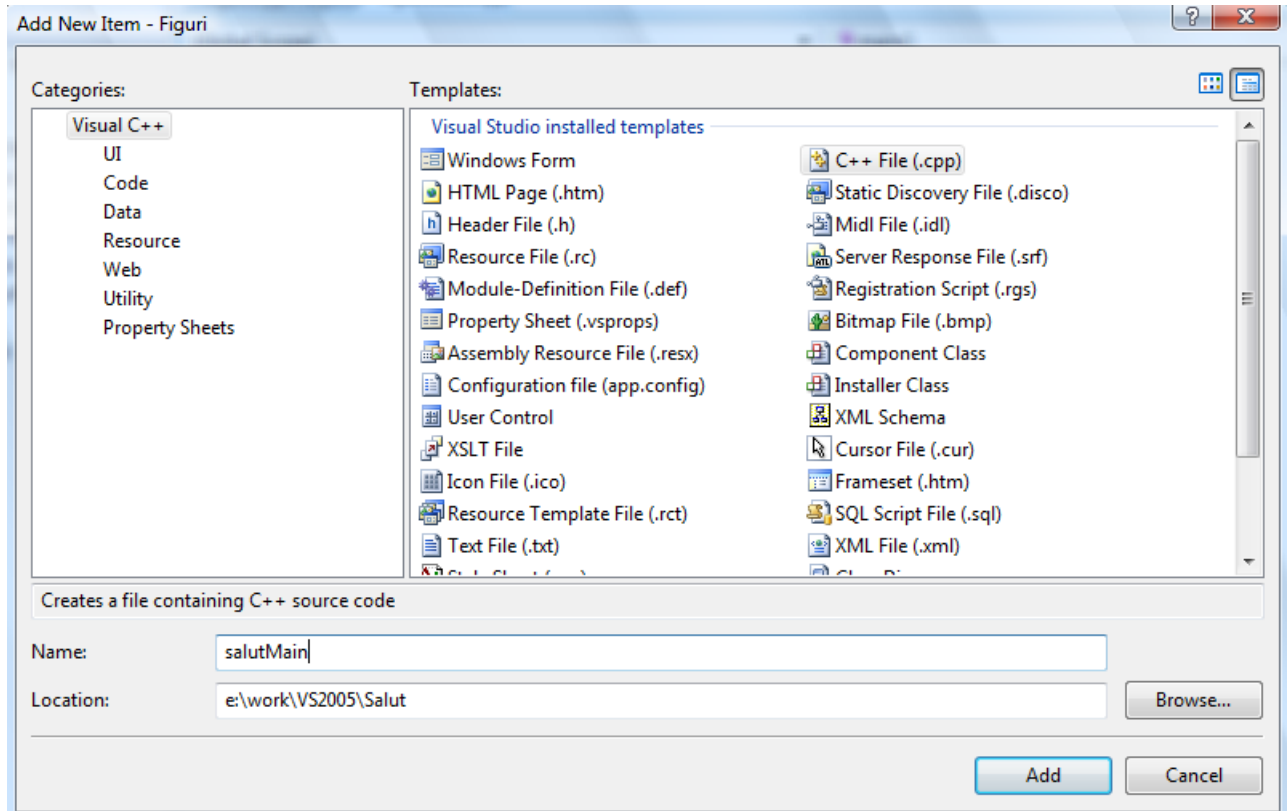


Fig. 6 Adăugare fișier sursă

2. Apare fereastra “Add New Item”.
3. În căsuța templates alegem *C++ File (.cpp)*
4. La name alegem numele fișierului. De obicei vom numi fișierul care conține funcția main() după regula *<numeProiect>Main*. În cazul nostru – *salutMain*:

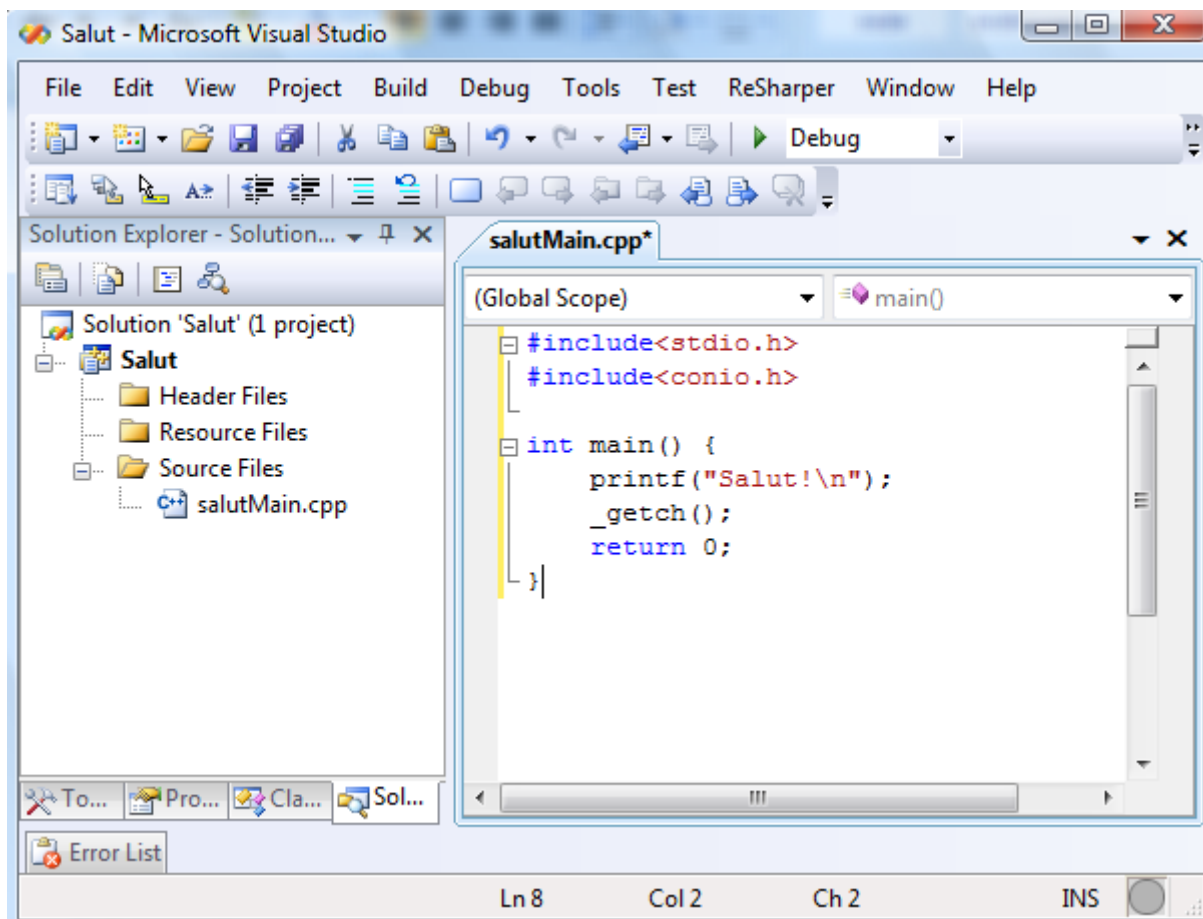


**Fig. 7 Adăugare fișier sursă – partea 2**

5. Apăsăm *Add*. Noul fișier este creat și deschis în editor.

### 3. Scrierea primului program

Să scriem un program simplu care afișează un mesaj la consolă, după care așteaptă apăsarea unei taste. Codul este prezentat în figura 8.



**Fig. 8 Scrierea codului sursă**

Observăm în dreptul numelui fișierului – salutMain.cpp, o steluță (\*). Aceasta înseamnă că fișierul conține modificări nesalvate. Liniile modificate sunt marcate cu galben în stânga editorului. Apăsăm **Ctrl+S** pentru a salva fișierul. Observăm cum steluța a dispărut.

Funcția `_getch()` la sfârșitul programului este echivalentul lui `getch()` din borland C. Ea așteaptă apăsarea unei taste fără să o afișeze pe ecran.

În Visual Studio codul este indentat în mod automat atunci când îl scriem. Chiar dacă programul nu este indentat așa cum ne dorim, putem să reformatăm codul în orice moment apăsând **Ctrl+A**, apoi **Ctrl+K**, **Ctrl+F**. Este o practică bună să avem codul tot timpul formatat. Formatarea ne ajută să înțelegem programul și să detectăm mai ușor erorile. În figura 9, se ilustrează modalitatea de formatare a codului:

1. În dreptul apelului `printf()` apăsăm tab pentru a-l indenta în mod corect la dreapta;
2. Apăsăm **Ctrl+A** pentru a selecta tot textul.
3. Apăsăm **Ctrl+K**, apoi **Ctrl+F**. Observăm cum codul este formatat înapoi în mod corect.

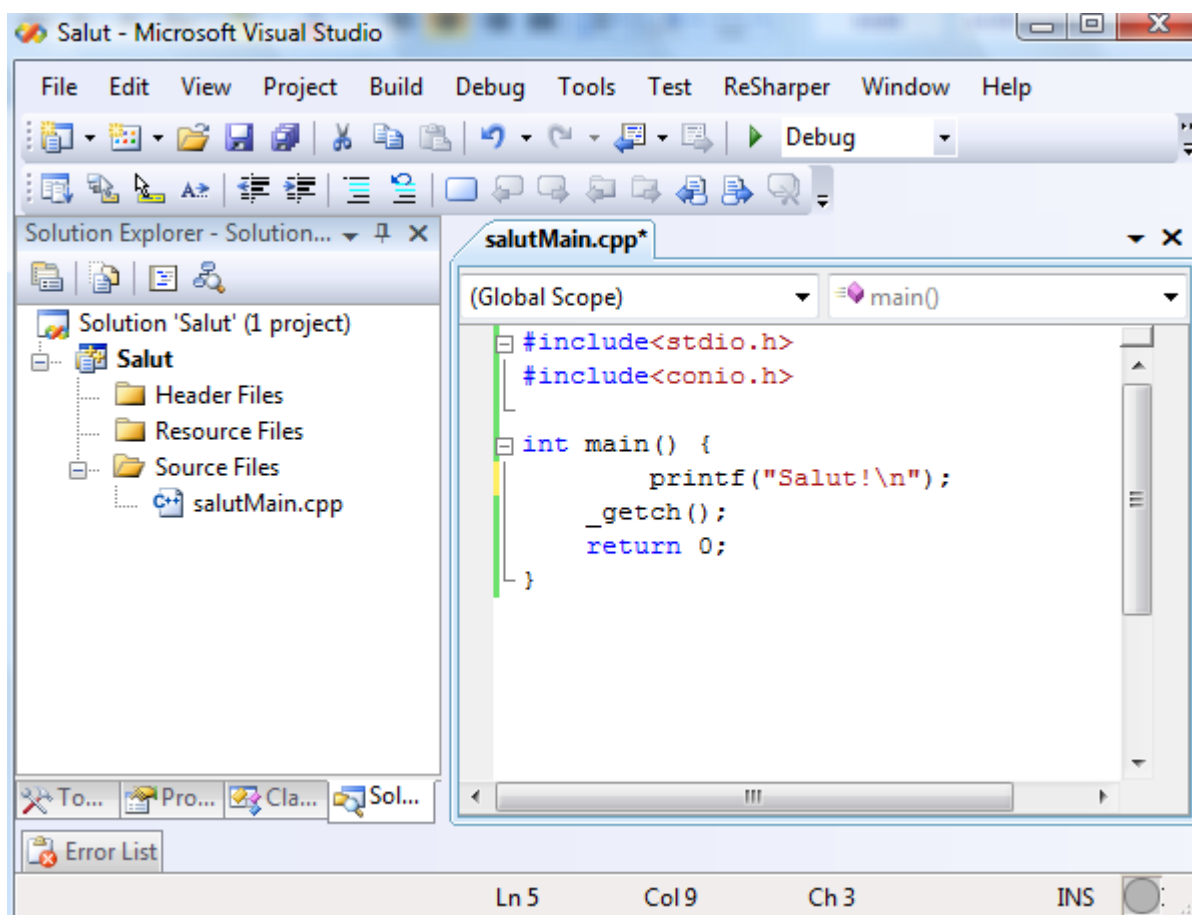


Fig. 9 Exemplu de cod neformatat

#### 4. Compilarea și rularea proiectului. Detectarea erorilor.

Pentru a compila proiectul apăsăm **Ctrl + Shift + B**, sau din meniul principal selectăm **Build** → **Build Solution**.

Dacă sunt detectate erori de compilare, în partea de jos a ecranului va apărea fereastra Error List. Să introducem intenționat o eroare pentru a vedea facilitățile acestei ferestre.

De exemplu, putem înlocui linia “return 0” cu “return x”. La compilare, compilatorul consideră că x este o variabilă nedefinită și afișează fereastra error list cu mesajul din figura 10.

Dacă dam dublu-click peste mesajul de eroare, cursorul se va deplasa la linia de cod conținând eroarea. Utilizarea unei variabile fără să fie declarată este una din cele mai comune erori de programare. În VS2008 aceasta poate fi detectată și corectată foarte ușor. Alte erori însă pot duce la mesaje de eroare mult mai obscure, și chiar linia de cod a erorii poate fi detetată greșit.

În general s-a observat următoarea regulă – eroarea trebuie căutată ori pe linia afișată în error list, ori cu o linie mai sus. De exemplu, ștergeți simbolul ; la sfârșitul uneia dintre linii. Veți vedea că mesajul de eroare ne duce cu o linie mai jos.

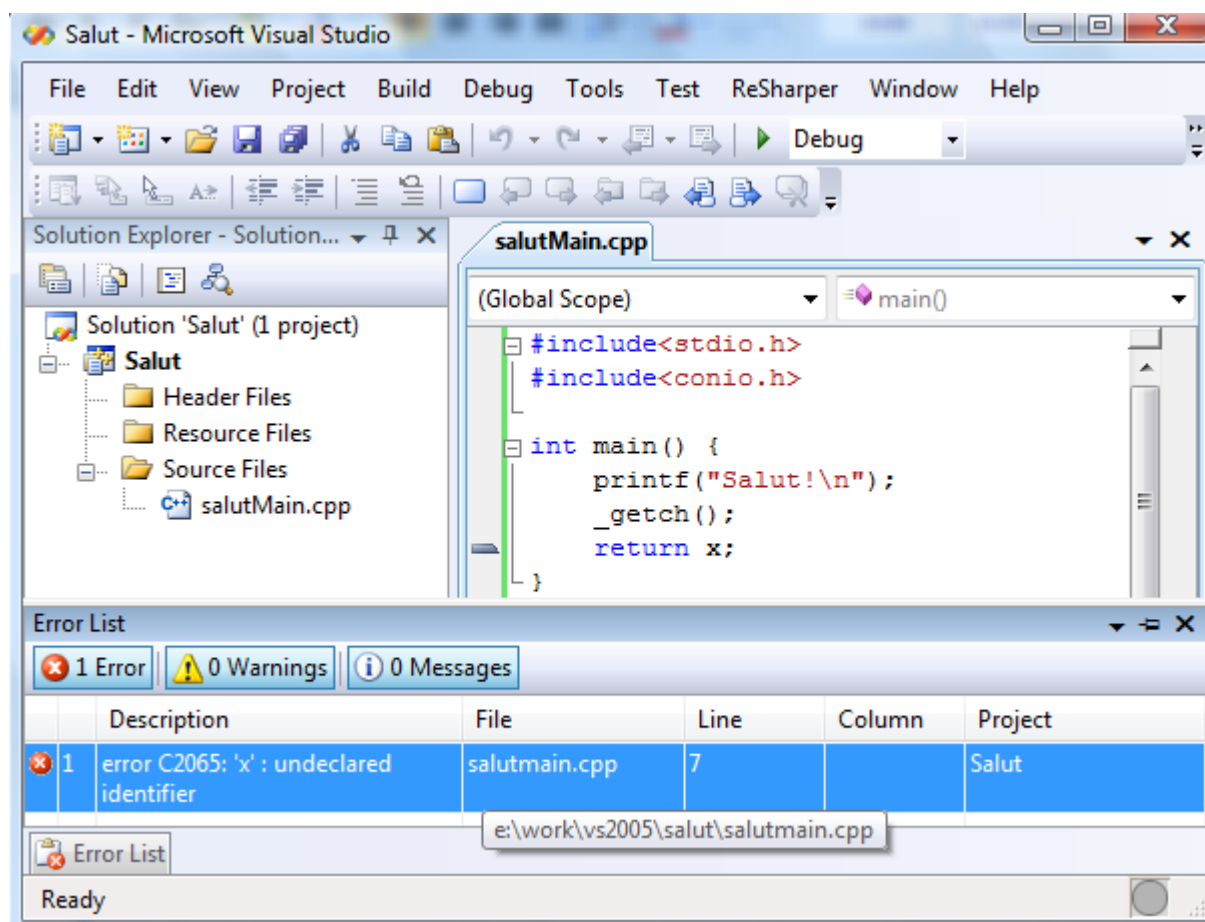


Fig. 10 Mesaj de eroare

Să corectăm programul și să îl compilăm din nou. Chiar dacă fereastra error list nu s-a ridicat după compilare, trebuie să o verificăm, pentru că aceasta nu se ridică atunci când programul are doar warning-uri. De cele mai multe ori warning-urile sunt și ele erori de program și trebuie să la eliminăm

Pentru a executa programul apăsăm **F5**. Se deschide o fereastră – consola în care rulează programul.

## 5. Debug

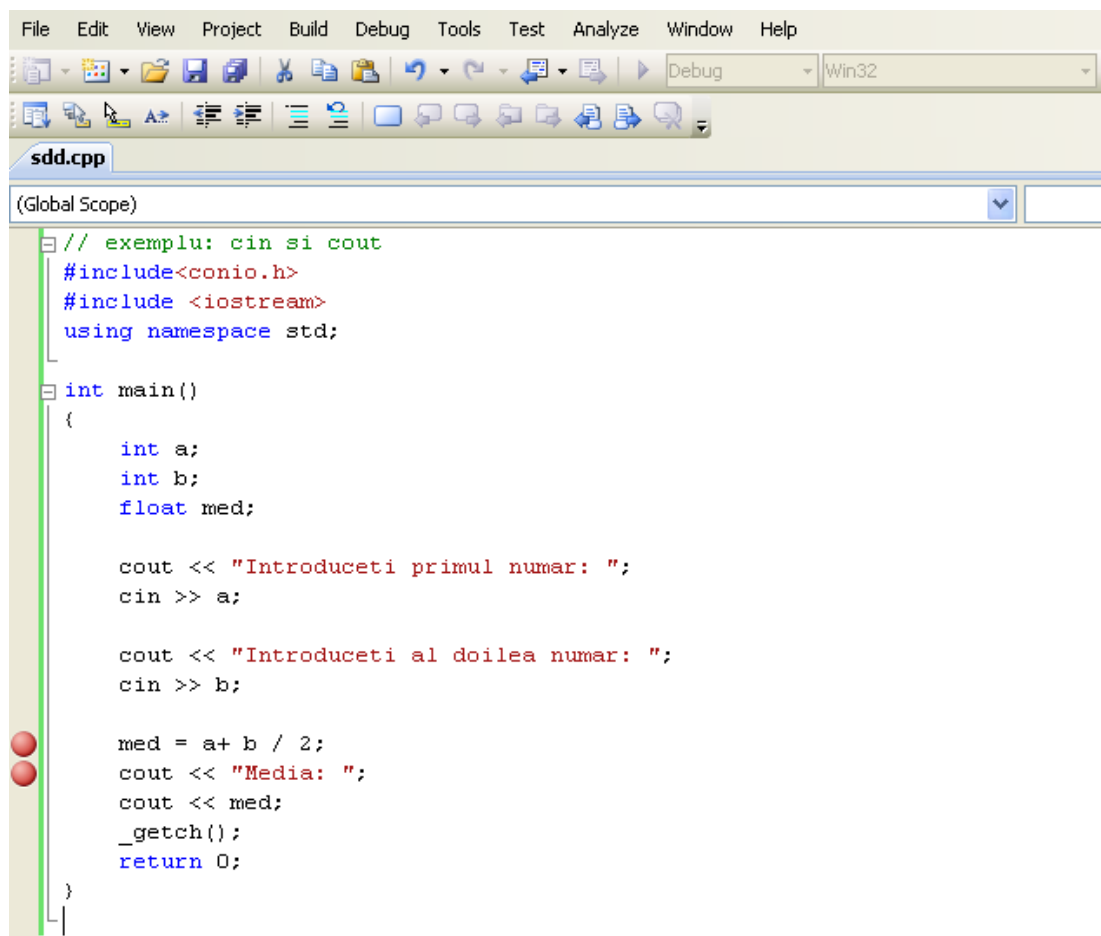
Debug este facilitatea oferită de mediile de dezvoltare de a analiza procesul de execuție a unui program în scopul de a detecta erorile. Este posibil de a rula programul instrucțiune cu instrucțiune, urmărind valoarea unor variabile după fiecare instrucțiune. Sau este posibil de a stabili anumite linii de cod la care dorim ca programul să se oprească, și să vizualizăm valoarea variabilelor alese de noi doar în acele puncte din program. Aceste linii în care dorim să intrerupem execuția programului se numesc puncte de oprire (breakpoint).

De exemplu, să considerăm programul de mai jos:



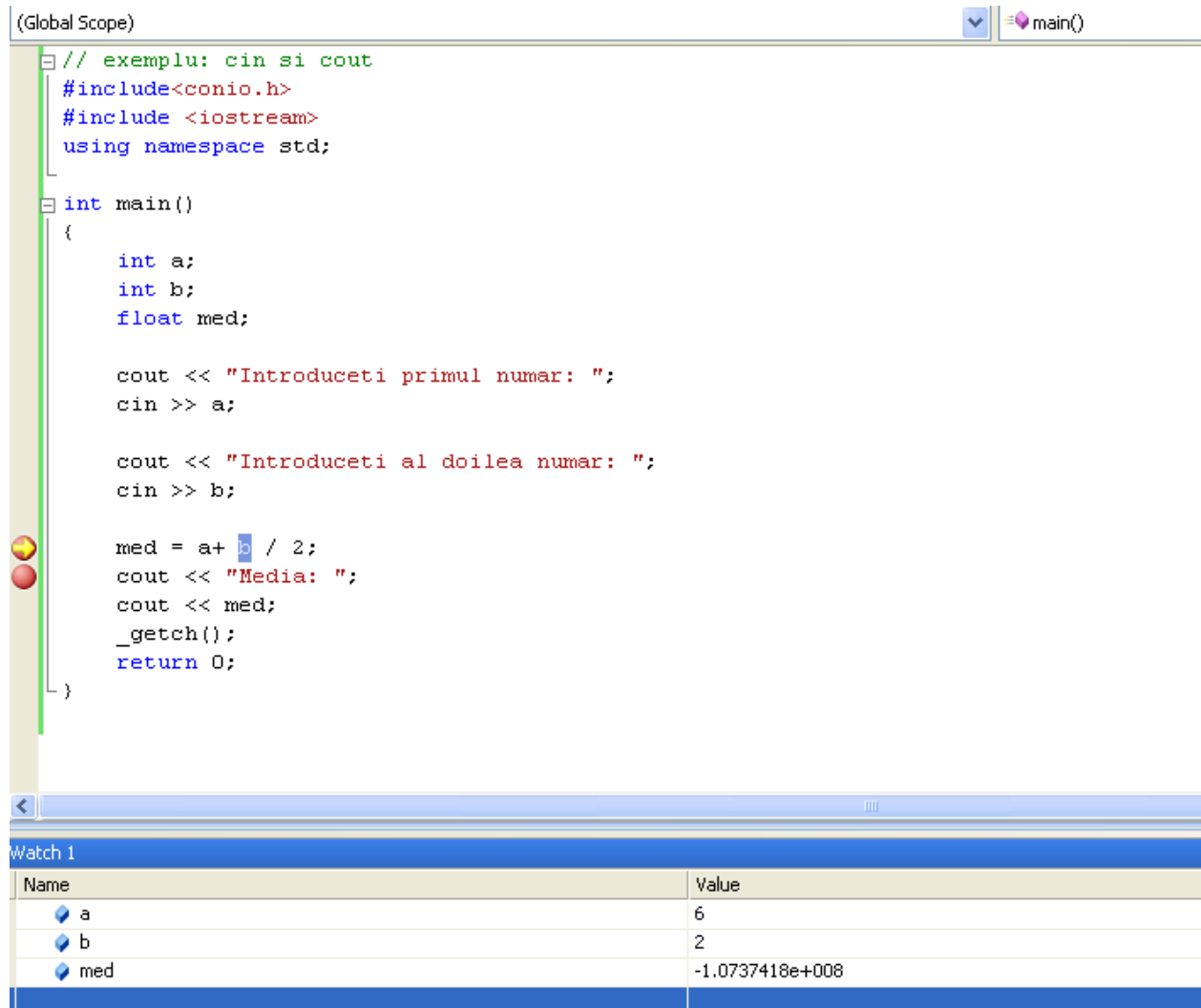
<pre> int main() {     int a;     int b;     float med;      cout &lt;&lt; "Introduceti primul numar: ";     cin &gt;&gt; a;      cout &lt;&lt; "Introduceti al doilea numar: ";     cin &gt;&gt; b;      med = a+ b / 2;     cout &lt;&lt; "Media: ";     cout &lt;&lt; med;     _getch();     return 0; } </pre>	<p>Introduceti primul numar: 6 Introduceti al doilea numar: 2 Media: 7</p>
--	--

vom dori să verificăm de ce media nu este 4 așa cum ne-am dori. În primul rând putem verifica dacă citirea a fost făcută corect, și în caz afirmativ, dacă operația de medie a fost făcută corect. Pentru aceasta, vom plasa 2 breakpoint-uri în funcția main() – unul după citirea celor două numere și unul după calcularea mediei. Pentru a plasa un breakpoint, ducem cursorul la linia de cod unde dorim breakpoint-ul, și apăsăm **F9**. Tot cu **F9** se poate scoate un breakpoint. În partea dreaptă a editorului de text va apărea o bulină roșie pentru fiecare breakpoint:



**Fig 11. Breakpoint-uri**

Acum apăsăm **F5** pentru a rula programul în mod debug. Vom observa că programul își începe execuția normal, ne cere datele de intrare, iar după ce le introducem consola se blochează. În acel moment ne întoarcem în Visual Studio și observăm că aranjamentul ferestrelor s-a schimbat, iar în dreptul primului breakpoint a apărut o săgeată:



**Fig 12. Oprirea execuției la un breakpoint**

Săgeata indică instrucțiunea la care s-a oprit execuția programului.

**Atentie!** Instrucțiunea la care se află săgeata încă nu s-a executat, dar instrucțiunea anterioară a fost executată!

Acum, putem să vizualizăm valoarea unor variabile. În mod evident, ne va interesa valoarea celor 2 variabile definite în `main()` – `a` și `b`. Pentru a le vedea, mergem în meniul principal → **Debug** → **Windows** → **Watch** → **Watch 1**.

Apare fereastra **Watch 1** în partea de jos. În această fereastră dăm click pe coloana **Name** și introducem numele variabilelor ce dorim să le vedem – întâi `a`, apăsăm enter, pe urmă `b`, enter:

În cea de-a 2-a coloană – **Value**, va fi afișată valoarea variabilelor, iar în cea de-a 3-a – **Type** – tipul.

Fereastra watch ne permite însă să vizualizăm nu doar variabile, ci și orice expresie. Putem să vizualizăm elementele unui vectorului, pe care VS va putea să le afișeze într-o formă mai relevantă.

Mai jos sunt sumarizate combinațiile de taste utile în VS.

Combinatia de taste	Efect
Ctrl + C	Copy
Ctrl + V	Paste
Ctrl + A	Select all
Ctrl + K, F	Format selected
Ctrl + A, K, F	Format all
Ctrl + Shift + B	Build all
F5	Debug, continue after breakpoint
F9	Insert / remove breakpoint

## 6. Documentație

Sursa recomandată de documentație este situl <http://www.cplusplus.com/> . În particular:

- <http://www.cplusplus.com/reference/library/> - documentația completă a tuturor header-elor din C, și a funcțiilor din ele. Vedeți meniul din stânga. Similar cu documentația din Borland C.
- <http://www.cplusplus.com/reference/iostream/> - documentația claselor ce realizează lucrul cu fișierele. Inclusiv obiectele cin și cout.
- <http://www.cplusplus.com/doc/tutorial/> - un material didactic alternativ despre limbajul C++.