

KAKO KOMPAJLIRATI?

Kratki vodič za sve koji žele naučiti kako pokrenuti vlastite programe na računalu.

Autor: Kristijan Franković

1. UVOD

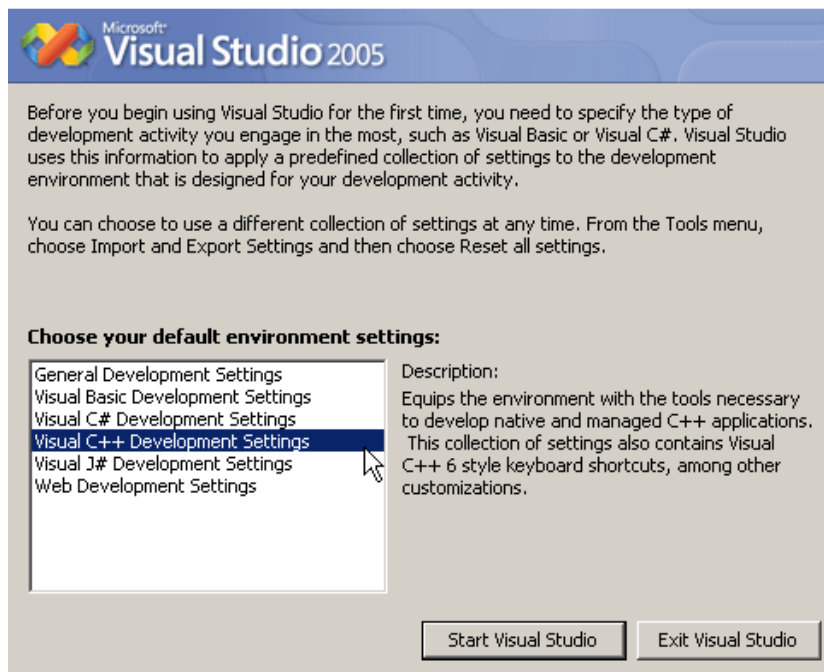
Bez velike količine vježbanja nema kvalitetnog znanja programiranja i programskog inženjerstva, a to je najlakše ostvariti radeći na računalu i prateći izlazne rezultate. U tu svrhu, mi ćemo pokazati kako se kompajlira (prevodi) program u dva vrlo popularna alata – Microsoft Visual Studio 2005 (dostupan putem MSDNAA), koji predstavlja punokrvni alat koji koriste milijuni programera diljem svijeta koji se bave razvojem ozbiljnih aplikacija, te mnogo lakši Dev C++, koji se nažalost više ne razvija, ali je vrlo lagan (tek oko 20 mb za razliku od Microsoftovog proizvoda koji teži nekoliko GB) a sadržava sve bitno za ugodan rad. Naravno, ovdje opisani alati ne predstavljaju jedine kompajlere za programski jezik C, ali su najpopularniji i oba su i više nego dovoljni za uspješno učenje programiranja.

2. VISUAL STUDIO 2005

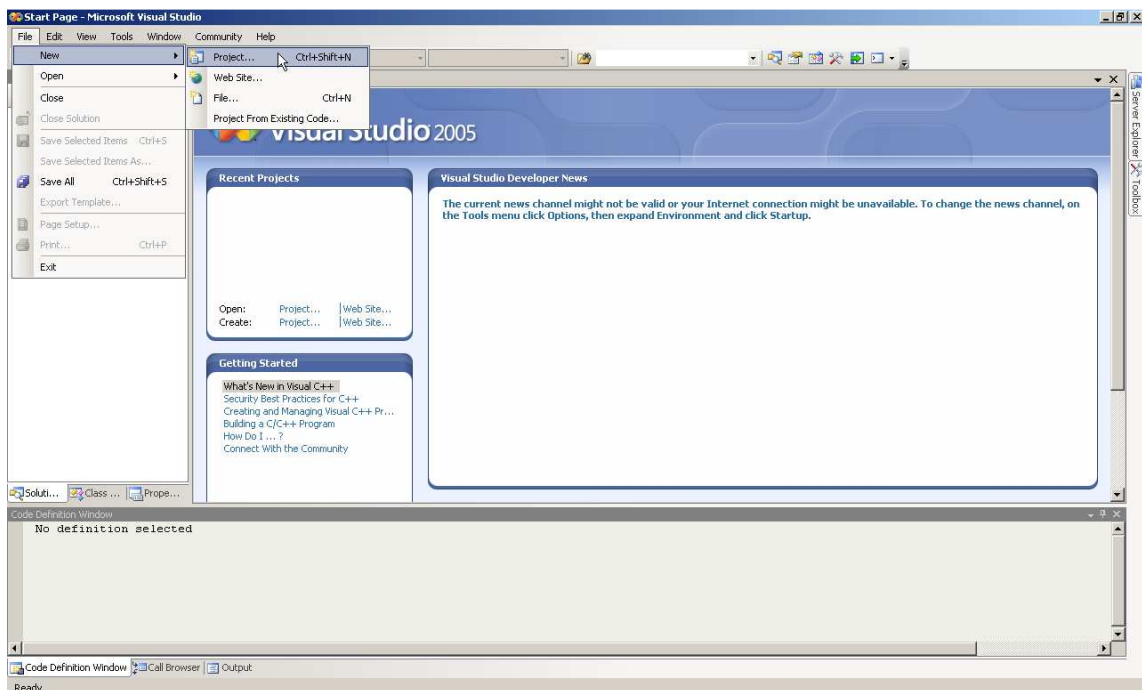
Proces instalacije, vjerujem nije potrebno detaljnije objašnjavati, stoga prijedimo na pokretanje samog programa. To činimo dvostrukim klikom na ikonicu prikazanu na slici niže:



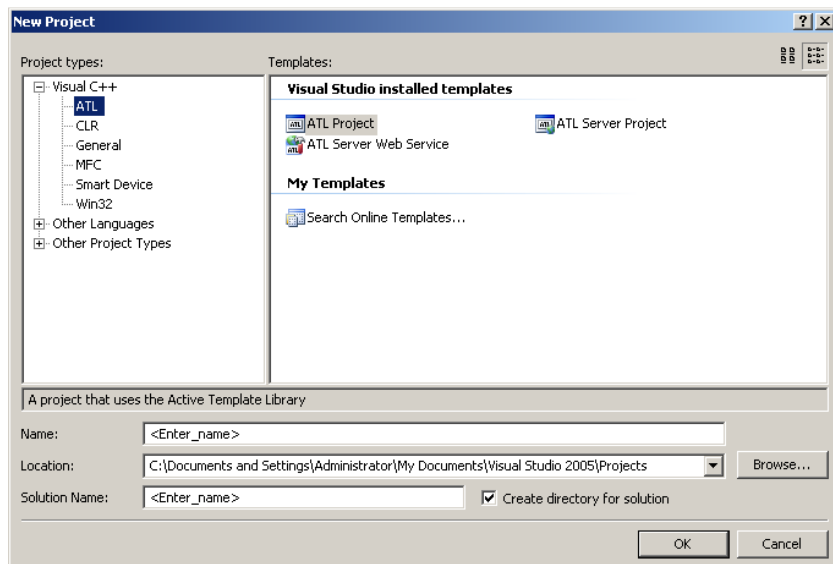
Nakon otprilike sekunde otvara nam se sljedeći okvir:



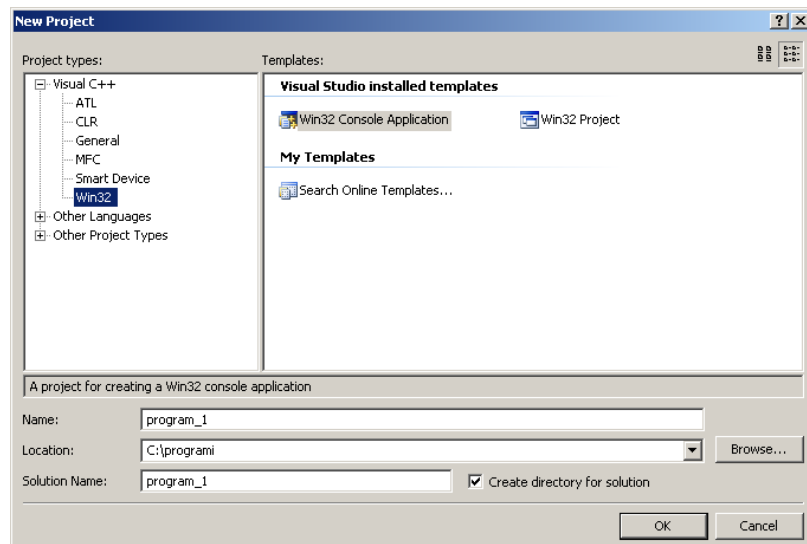
Ovdje odabiremo Visual C++ Development Settings i pritišćemo tipku Start Visual Studio. Nakon još koje sekunde otvara nam se razvojno sučelje ovoga Microsoftova alata prilagođenog za programiranje u programskom jeziku C++. Sučelje je prikazano na idućoj slici:



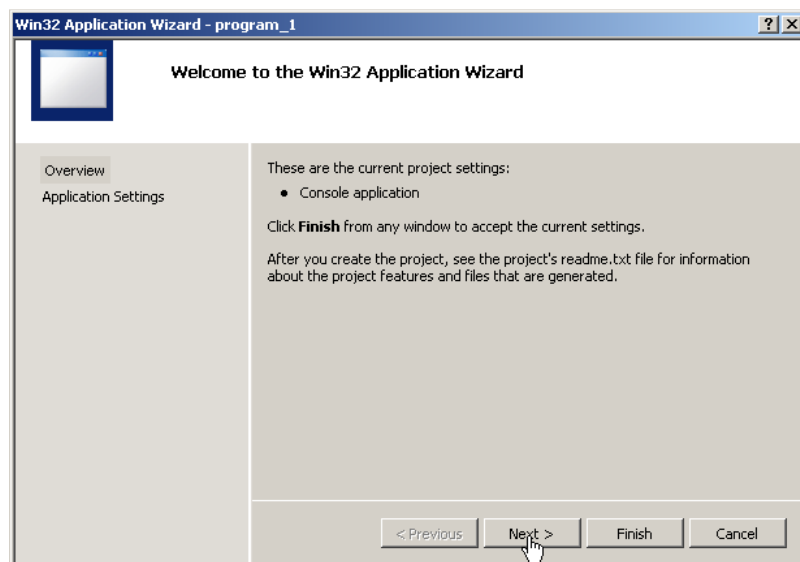
Kada nam se otvori sučelje kao na gornjoj slici, odabrat ćemo izbornik File i u njemu podizbornik New, te pritisnuti na opciju Project (kao što je vidljivo na slici). Potom nam se otvara slijedeći prozor:



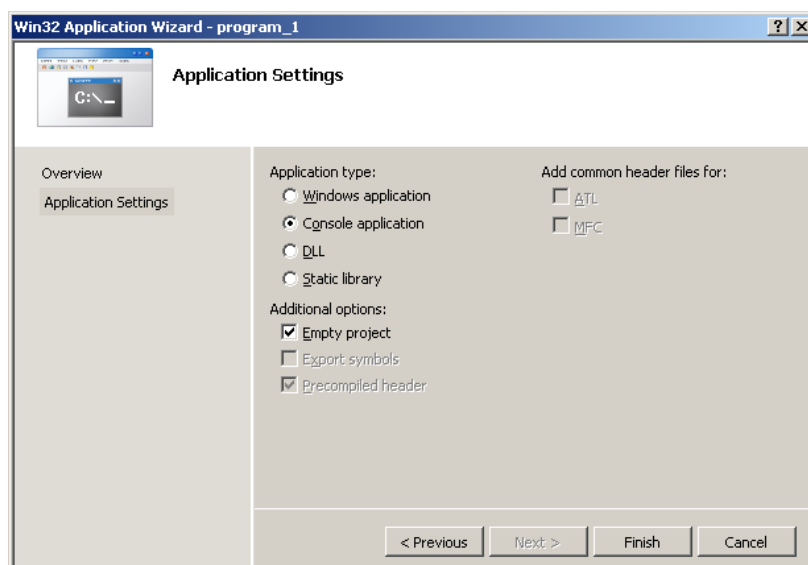
U velikom lijevom okviru pod „Project types“ odaberemo pod opcijom Visual C++ opciju Win32. Potom desno kliknemo na Win 32 Console Application. Dolje u okvir imena name možemo upisati ime koje želimo, kao i pod solution name. Location je mjesto gdje će se nalaziti naš program. Kod mene to primjerice izgleda ovako:



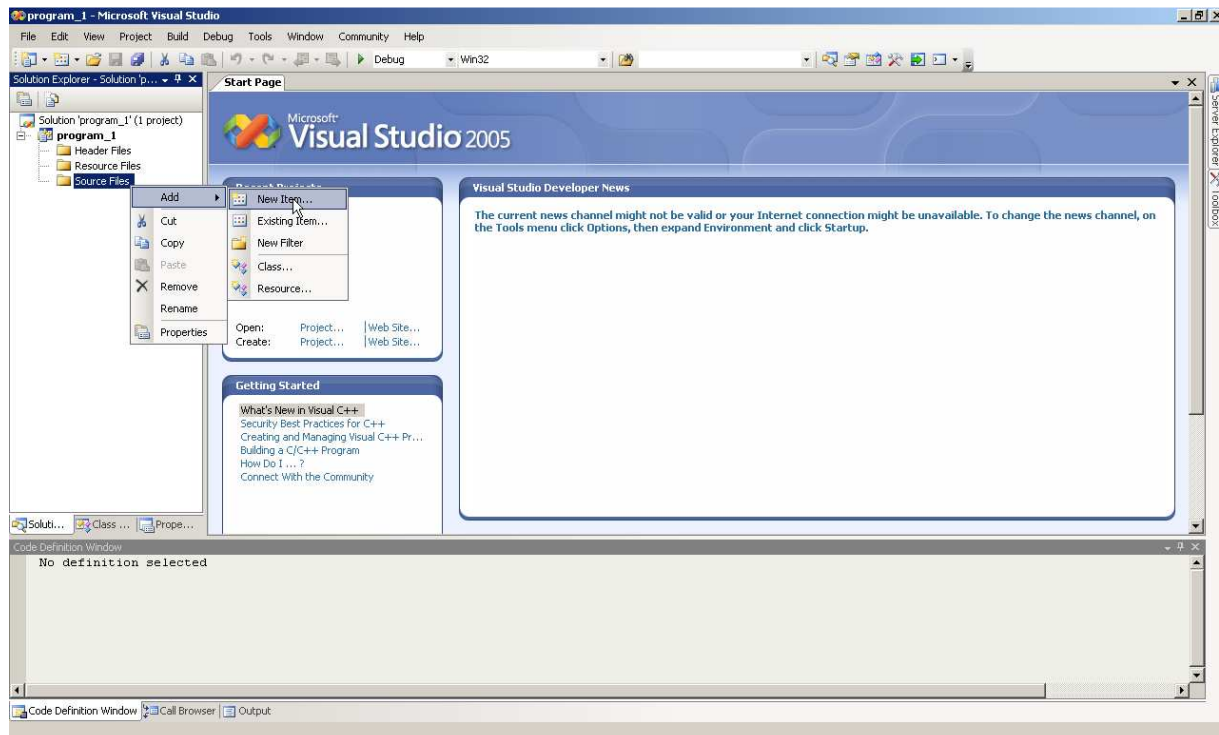
Potom odabiremo tipku OK. Ona nas vodi na čarobnjak za izradu projekta. Prvi ekran izgleda ovako:



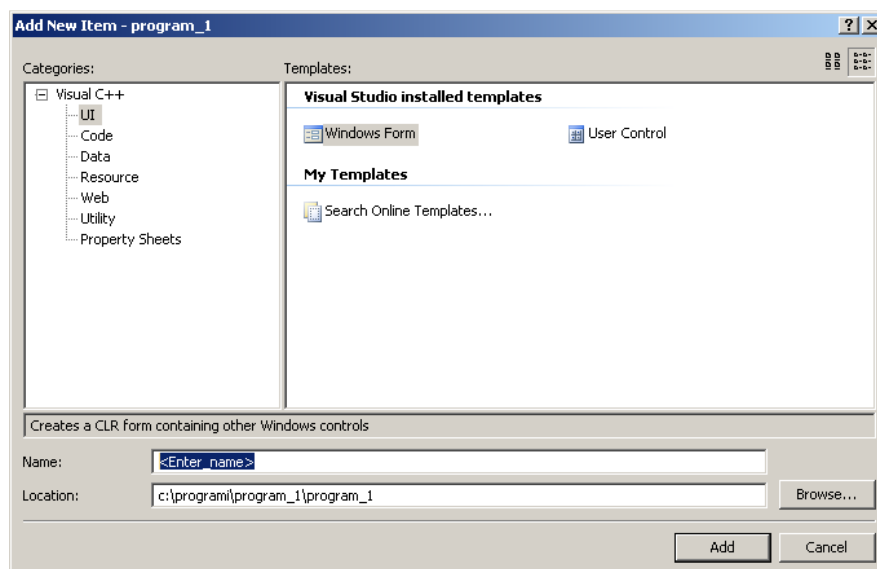
Ovdje samo odaberemo next. To nas odvede na idući ekran:



Ovdje pod Application type mora biti obavezno odabran opcija „Console application“. Pod additional options stavimo kvačicu na Empty project, odnosno prije pritiska na tipku Finish ekran mora izgledati kao na prethodnoj slici. Nakon toga čarobnjak se zatvara, a otvara se naš novi projekt u Visual Studiju. Samo sučelje mora izgledati otprilike ovako:

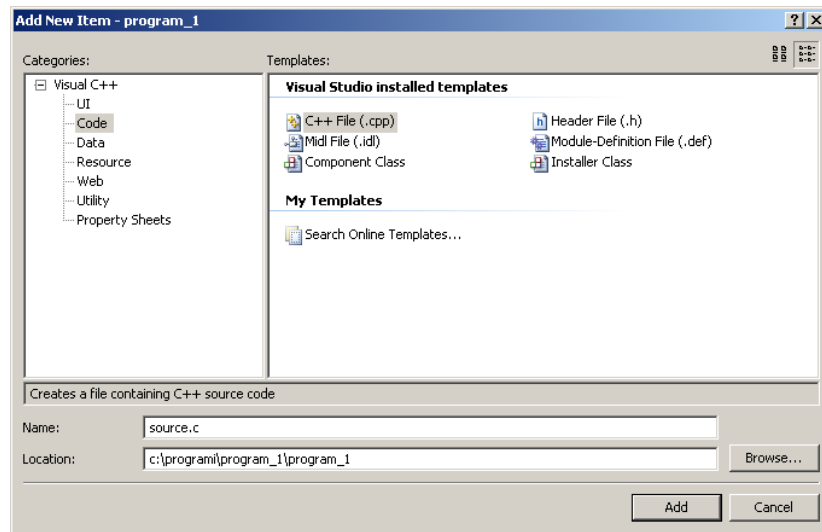


Lijevo se nalazi izbornik Solution explorer. U njemu je vidljiv naš novi projekt. Imamo tri mape, Header files, Resource files i zadnja je Source files. Desnim klikom na source files otvara nam se novi izbornik u kojem biramo podizbornik Add a u njemu odabiremo opciju New Item, kao što je vidljivo na našoj slici. To otvara sljedeći okvir:

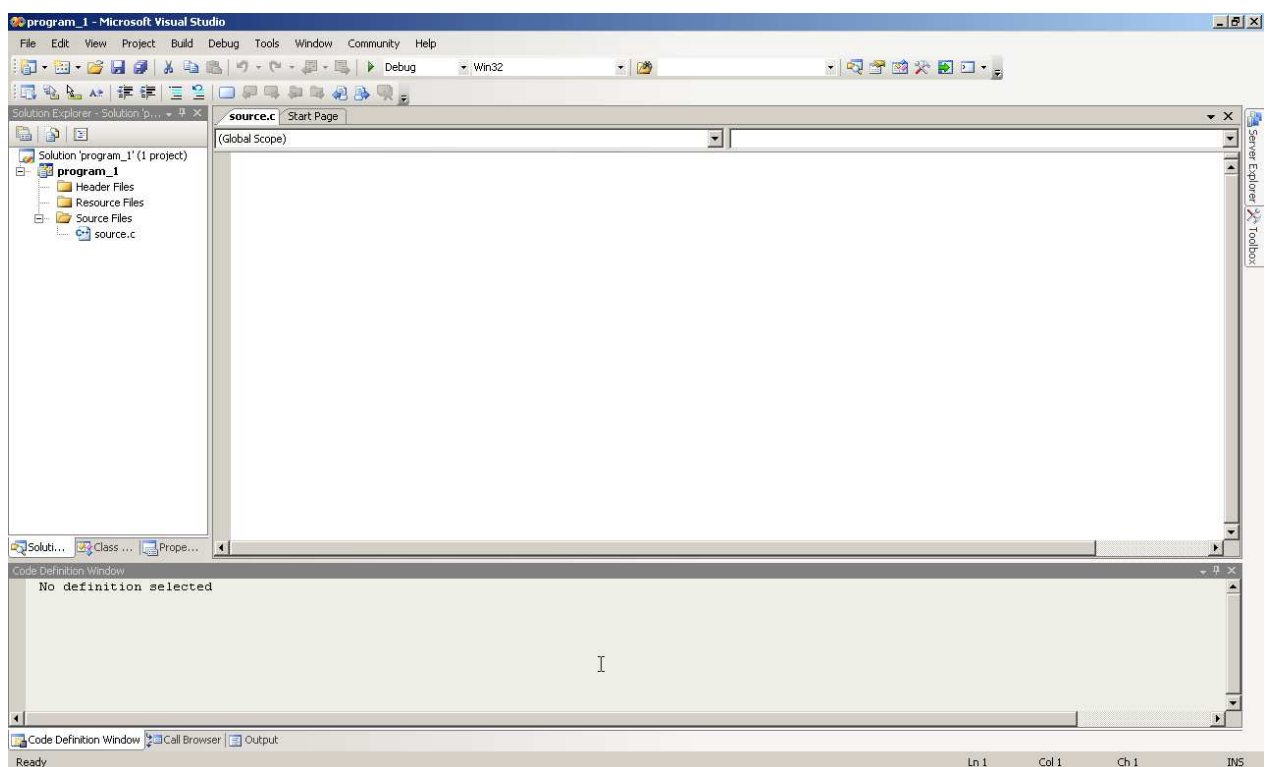


Ovdje je pod Categories potrebno odabrati Code i potom desno pod templates C++ file (.cpp). Name obavezno upisujemo neko ime. Ja osobno volim dati ime „source.c“. Sam taj file će biti smješten u onu mapu koju smo odabrali prilikom stvaranja projekta što nam je i vidljivo pod location. Primjetite da sam nakon imena stavio „c“ (source.c). ovo je dosta bitna stvar, jer u biti ovdje se radi o C++ razvojn

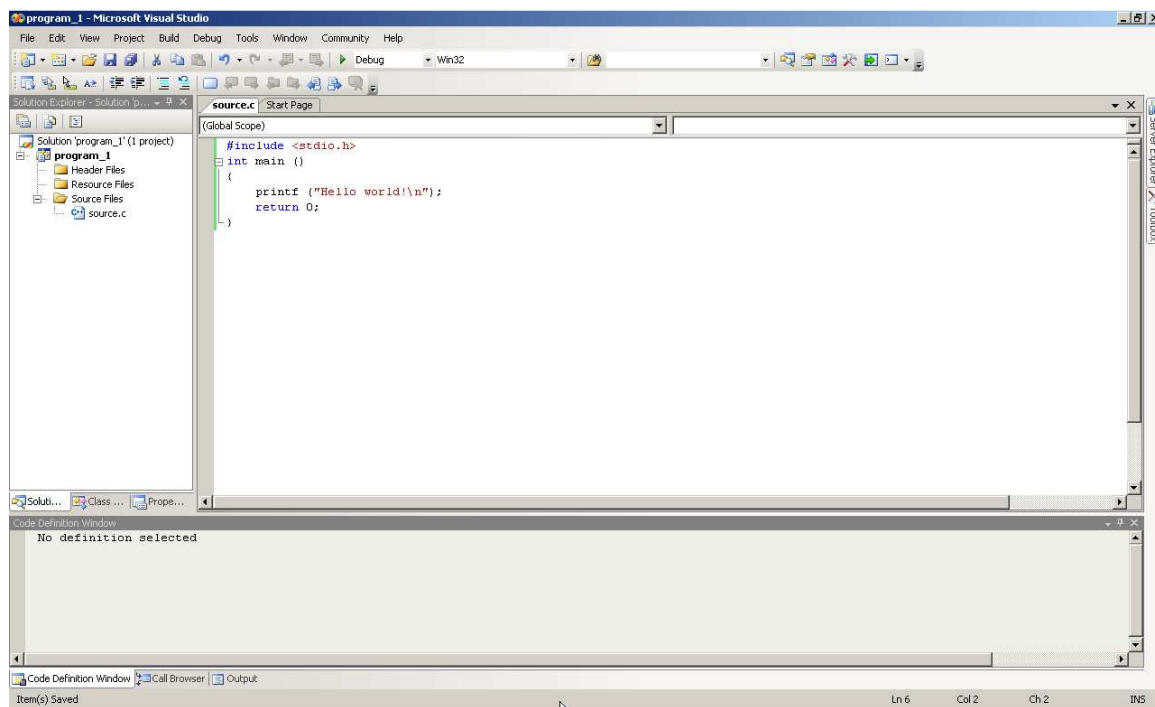
sučelju koje će ako ne stavite .c na kraj imena vašeg koda automatski staviti .cpp i raditi će kao da se radi o programskom jeziku C++. Iako u praksi neće stvarati probleme, ipak radije stavite .c na kraj. Dakle ispravan okvir prikazan je na slijedećoj slici:



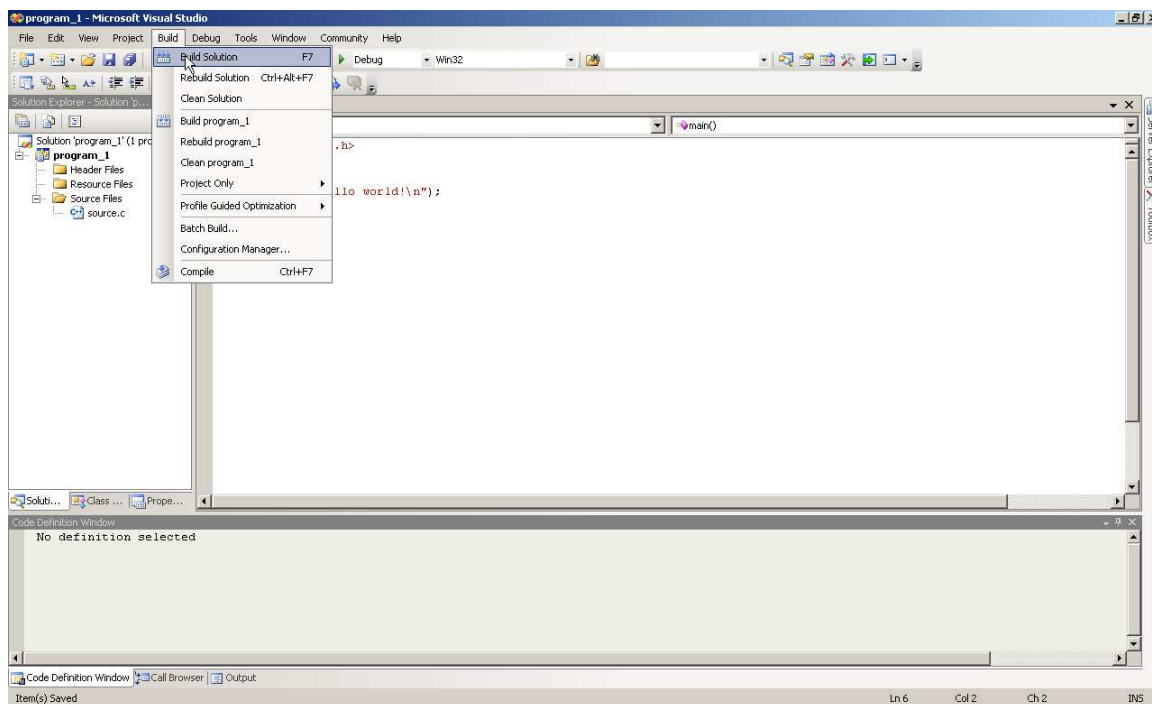
Odaberemo tipku add i ponovo se vraćamo u razvojno sučelje koje sada izgleda ovako nekako:



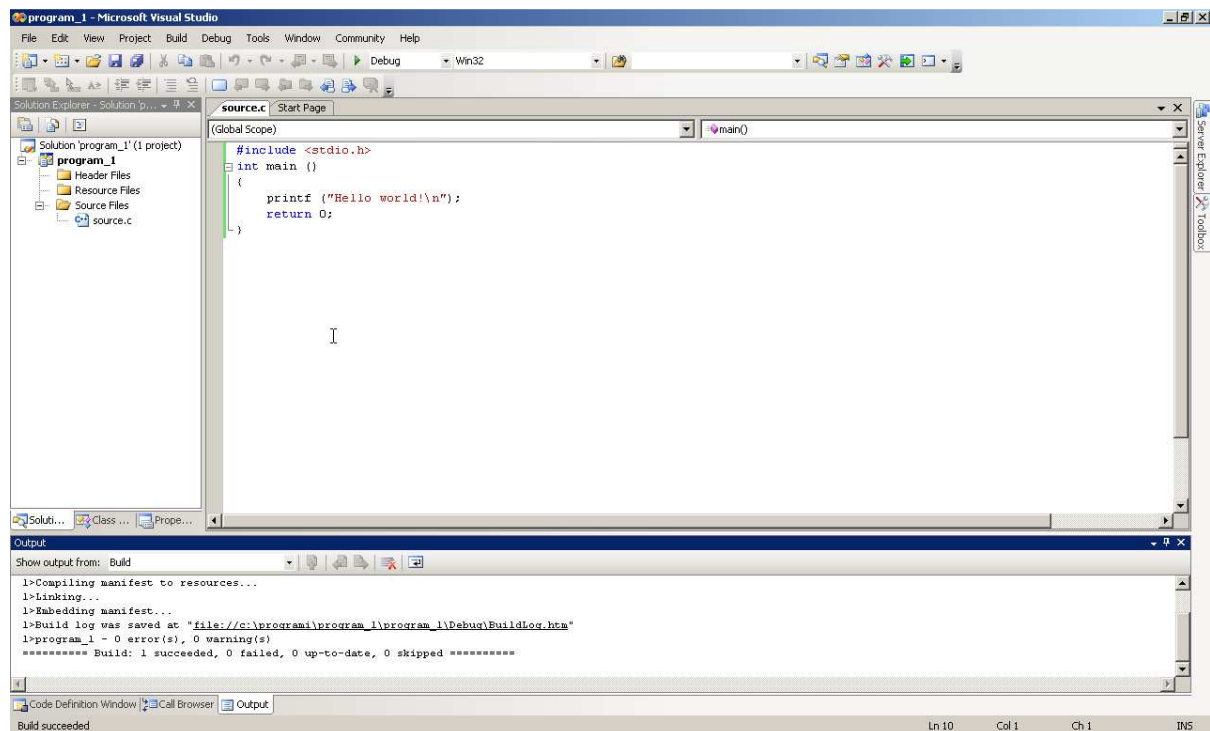
Dakle lijevo imamo Solution Explorer, desno od toga upisujemo kod. Ispod toga ovo veliko bijelo „Code definition Window“ na služi za praćenje eventualnih grešaka prilikom pokretanja koda. Stoga, idemo napisati i pokrenuti naš prvi program.



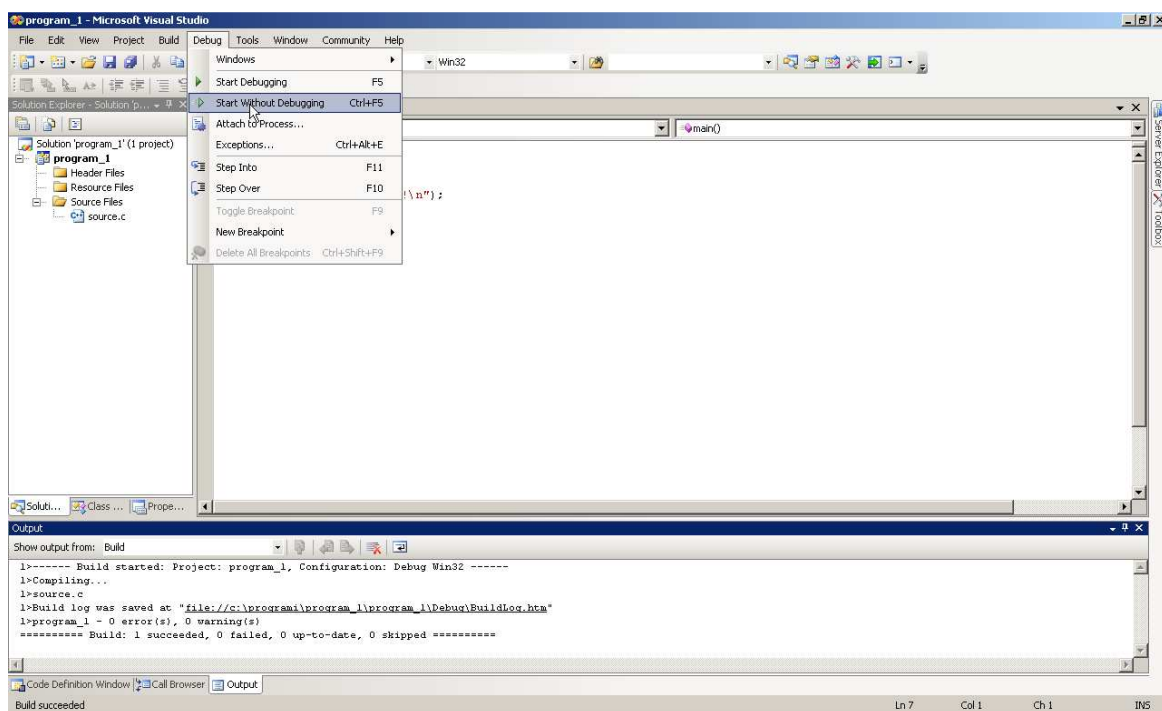
Evo, upisali smo poznati „Hello World“ program. Uz rub je vidljiva mala zelena traka. Ona označava ono što ste pohranili. Pritiskom na svima poznatu disketicu (Save). Ukoliko niste ništa pohranili traka će biti žuta, a ukoliko ste samo dio dodali ili mijenjali, taj dio koda će požutjeti. Klikom na disketicu (odnosno save) pohraniti ćemo kod na disk i traka će pozelenjeti. Sada je vrijeme da pokrenemo program i provjerimo ispravnost našeg koda. To činimo tako da u alatnoj traci pri vrhu odaberemo izbornik Build, a u tom padajućem izborniku odaberemo Build solution:



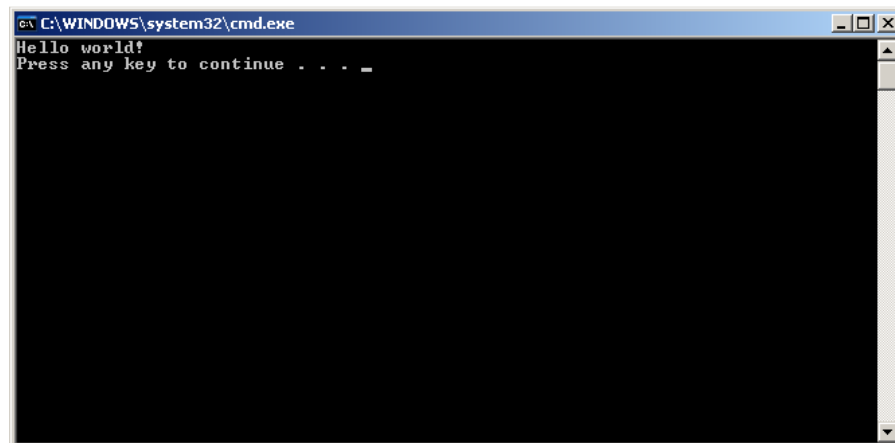
Potom on krene sa „buildanjem“. U donjem okviru (koji poplavi) se počni ispisivati neke poruke. Situacija bi otprilike ovako nekako trebala izgledati:



U tom donjem okviru vidimo jednu zanimljivu poruku. 0 error(s), 0 warning(s). Odlično, to znači da nemamo pogrešaka. Ukoliko uz error(s) stoji neka druga brojka osim 0 to znači da imamo nekakvu sintaktičku pogrešku u programu. Tada nažalost nećemo moći naš program pokrenuti te će biti potrebno pronaći te pogreške i ispraviti ih. Ukoliko imamo nekakve warninge, program će se moći pokrenuti, međutim Visual Studio nam javlja da smo nešto mogli i „bolje napisati“, odnosno da možda nešto neće raditi dobro. Ali u većini slučajeva će ipak raditi. Kako god, idemo sada napokon pokrenuti naš program. Sad ćemo odabrati izbornik Debug i u njemu opciju Start Without Debugging.



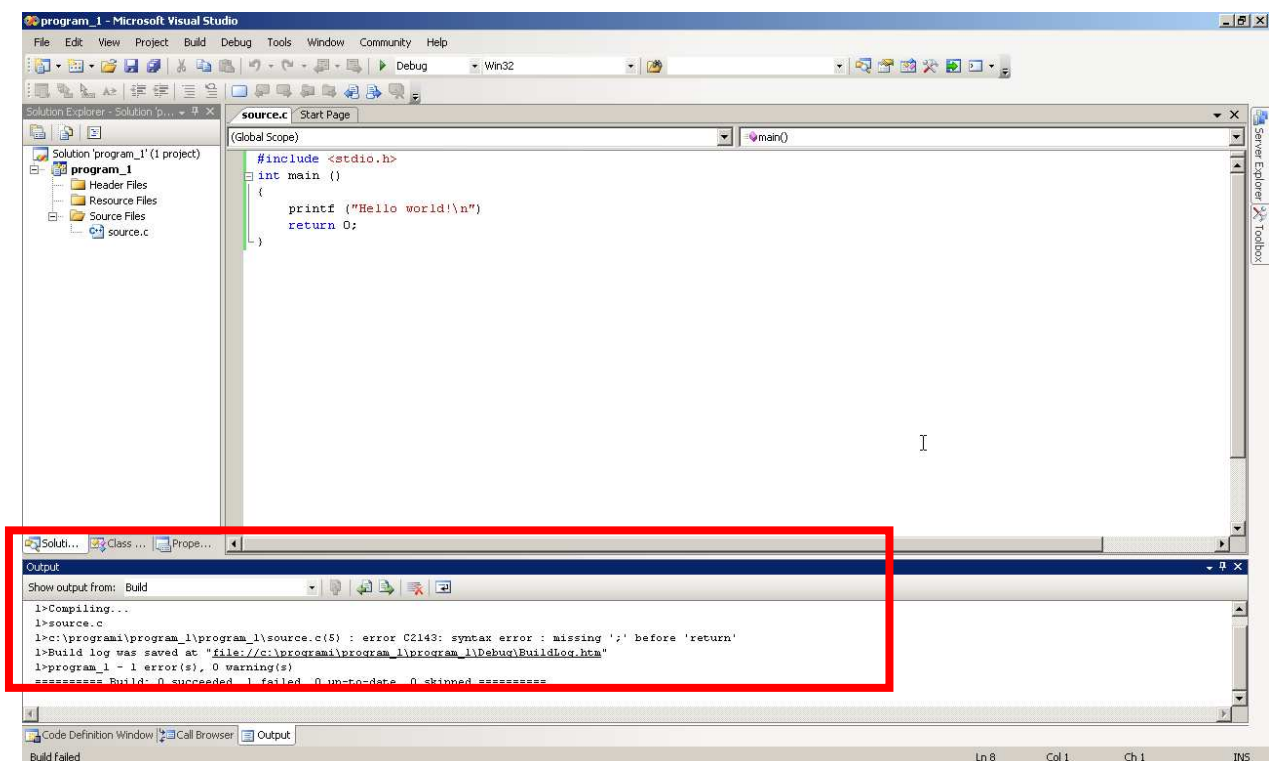
Nakon toga naš se program pokreće! Kako to izgleda, vidljivo je na sljedećoj slici:



Nadivimo se našem programerskom umijeću i potom zatvorimo pritiskom na X. Vraćamo se u naše razvojno sučelje. Svaki puta kada nešto u kodu promijenimo potrebno je prvo pohraniti ono što smo mijenjali, potom iz izbornika Build odabrati Build (može i rebuild solution), te potom iz izbornika Debug odabrati Start Without Debugging (može i CTRL+F5 naravno). Idemo sada malo promijeniti. Ukloniti ćemo na kraju retka gdje ispisujemo „;“ čime ćemo izazvati sintaktičku pogrešku:

```
#include <stdio.h>
int main ()
{
    printf ("Hello world!\n")
    return 0;
}
```

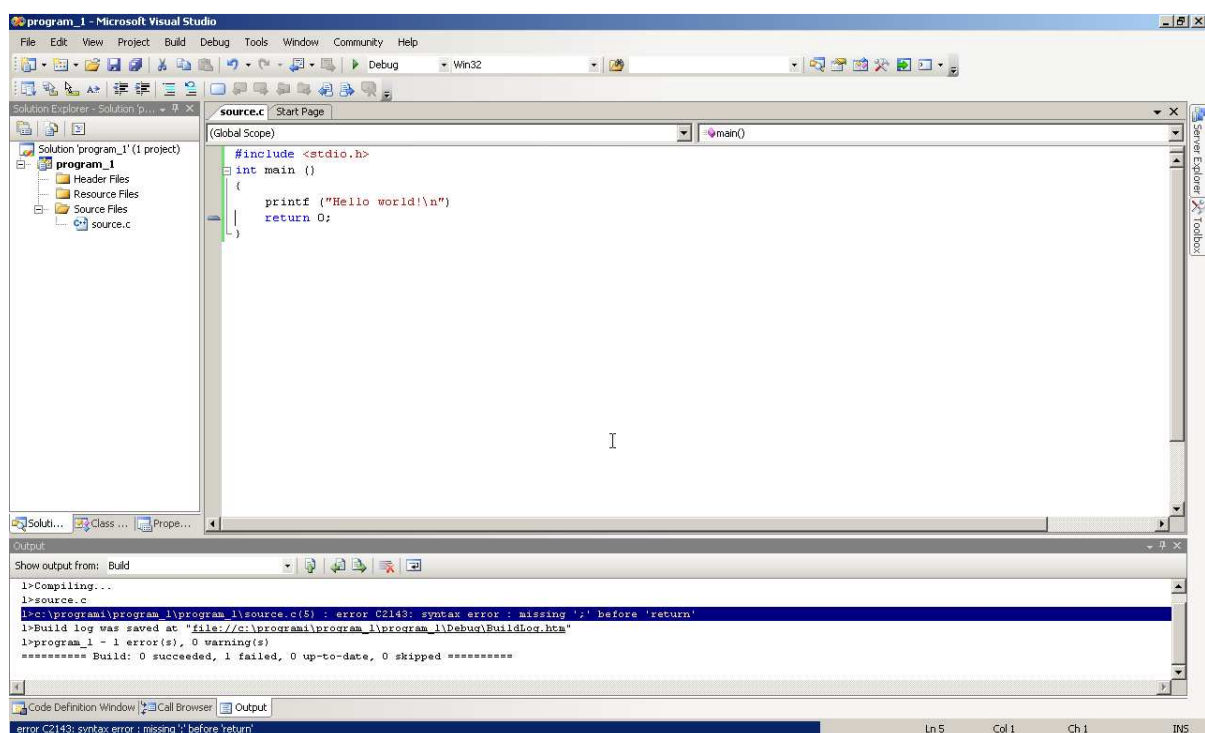
Vidimo da je taj dio koda požutio, što znači da smo mijenjali a nismo spremili. Spremimo kod. Potom odaberemo Build -> Build solution. On svoje izvrši i gle čuda, u donjem okviru se ponovo ispiše nešto:



Evo, okvir smo i uokvirili da se bolje vidi na što mislim. Pogledajmo ga uvećanog:

```
Output
Show output from: Build
l>Compiling...
l>source.c
l>c:\program1\program_1\program_1\source.c(5) : error C2143: syntax error : missing ';' before 'return'
l>Build log was saved at "file://c:\program1\program_1\program_1\Debug\BuildLog.htm"
l>program_1 - 1 error(s), 0 warning(s)
===== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped =====
```

Krasno – imamo error. Ako pogledate iznad ovog brojača pogrešaka, imamo točan opis greške error C2143: syntax error : missing ';' before 'return' . Ako dvostruko kliknemo na taj redak, u gornjem prozoru gdje je kod će nam se čak pojaviti mala strelica koja će nam reći gdje je ta greška. Ako znate imalo engleskog, razumjet ćete što nedostaje:



Ovdje nam on kaže – nedostaje „;“ prije return-a. Strelica je kao što vidimo na return. I stvarno, prije returna nedostaje ;. Pa stavimo ju onda!

```
#include <stdio.h>
int main ()
{
    printf ("Hello world!\n");
    return 0;
}
```

Evo, stavili smo ; (ovdje sam je označio plavom bojom). Sada kao što već znamo, prvo buildamo i potom odaberemo „Run Without Debugging“. I naš program se pokreće. Naravno, ovdje će vam se nemali broj puta dešavati da će te imati 20 – 30 errora i warninga. Ispravite ih jednog po jednog. Ništa strašno, dešavalo se i autoru ove knjige.

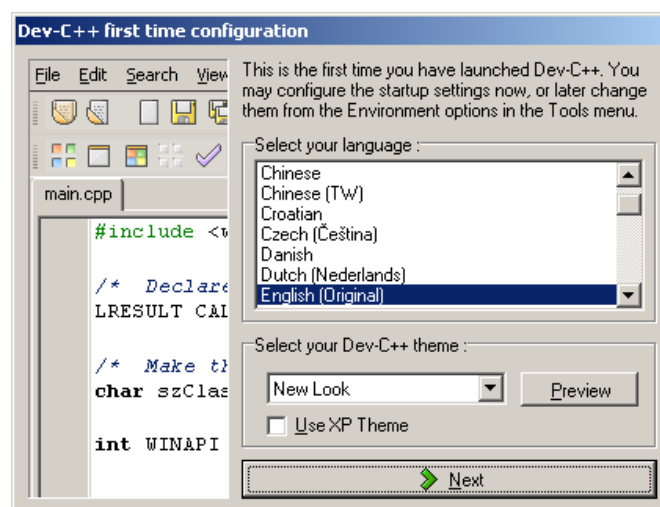
Sad se mnogi sigurno pitaju – „pa dobro, može li jednostavnije?“ Odgovor je – može. Pogledajte iduće poglavlje.

3. DEV C++

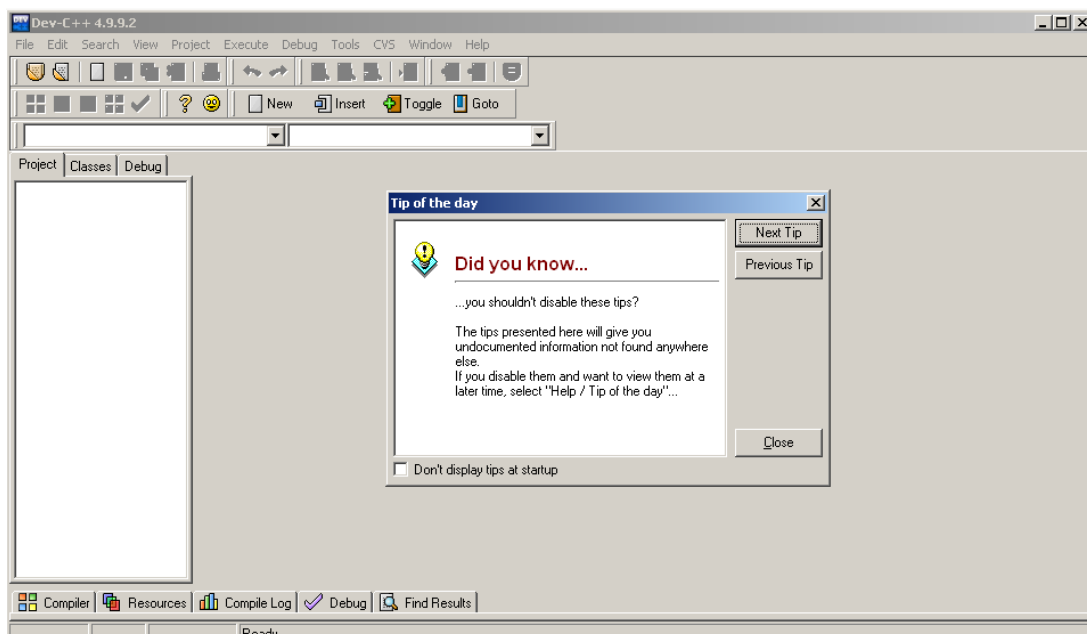
Instalacija ovog alata kao i u slučaju prethodno navedenog također neće predstavljati problem, a i vaš tvrdi disk će biti mnogo sretniji sa ovim programom u odnosu na Microsoftov. Stoga, idemo ga pokrenuti! Pokrećemo ga klikom na slijedeću ikonicu:



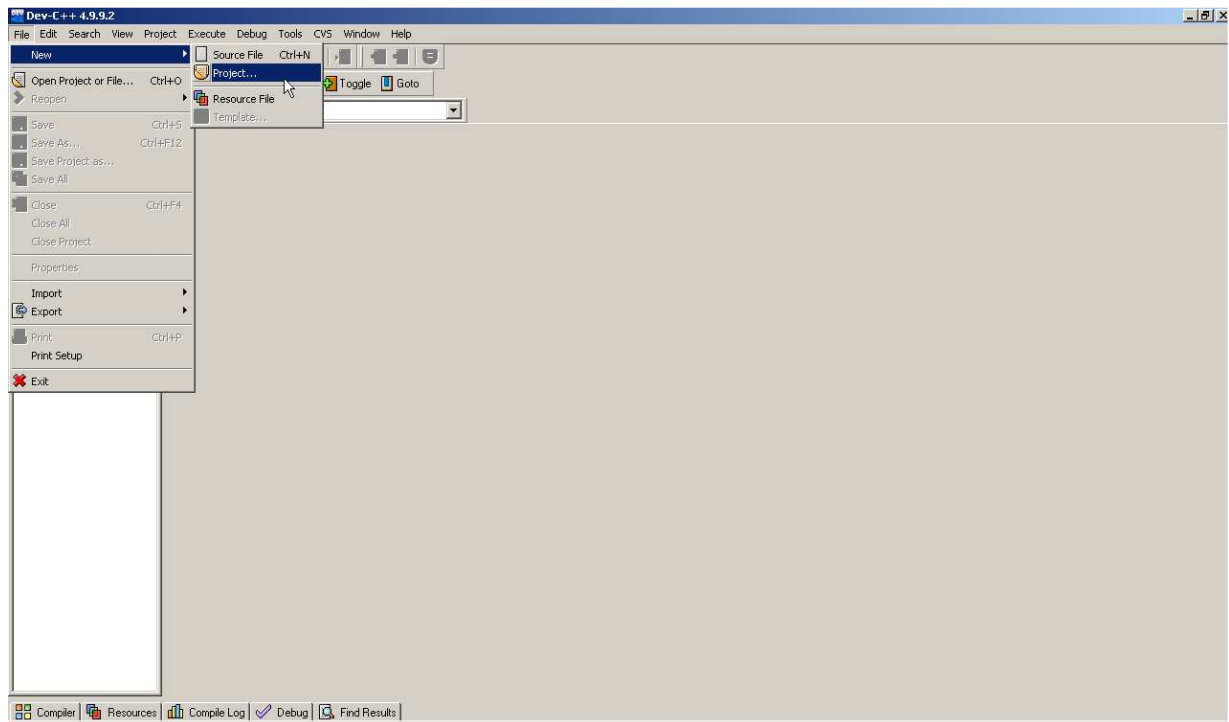
U mom slučaju, otvorio se okvir u kojem se nudi sam jedna tipka (OK) i koja mi govori da se radi o beti i o još nekim informacijama. Pročitajte što piše i kad vam već nudi taj jedan gumb, pritisnite ga. Otvara vam se već nešto korisniji prozor – Dev-C++ first time configuration.



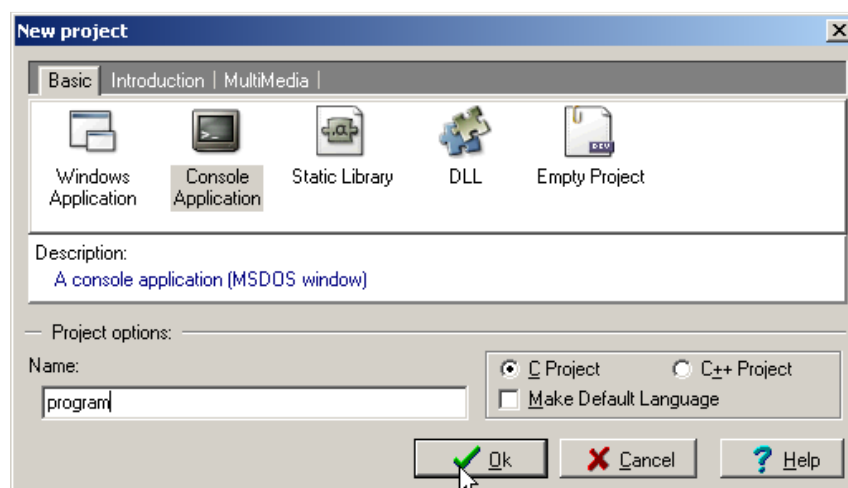
Odaberite jezik koji želite koristiti, (ja sam recimo ostavio engleski, unatoč tome što nudi hrvatski). Pritisnite next. Otvara se novi prozor na kojem vas pita želite li koristiti svojevrstu pomoć. Ukoliko ne radite na 286-ici slobodno ostavite „Yes...” i pritisnite Next. Otvara se novi okvir. I ovdje slobodno kliknite Next, nema smisla da se zamarate sa takvih tehnikalijama u ovom trenutku. Potom sam program par sekundi obavlja neke operacije. Nakon što završi ispiše poruku i vi odaberete OK, što je opet jedini gumb koji vam nudi. Konačno se otvara razvojna okolina samog alata, što otprilike izgleda ovako:



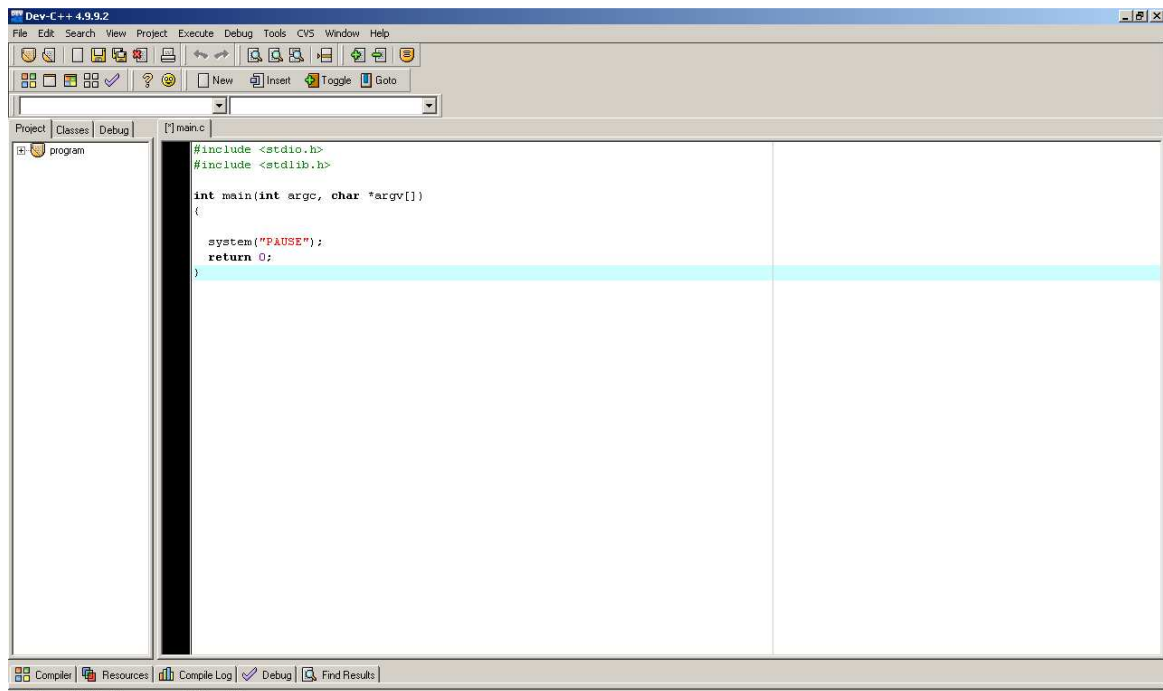
Sa njime se otvorio i dosadni „tip oft he day“. Iako vas u samom tekstu pokušava nagovoriti da ga ne ugasite, vi ipak odaberite opciju „Don't display tips at startup“ i odaberite Close. Odlično, sad napokon možemo ozbiljno raditi.



Na alatnoj traci odaberemo File, potom podizbornik New i nakon toga odaberemo opciju Project. Otvara nam se okvir kakav vidimo na slijedećoj slici:



Pod Basic odaberemo „Console application“, lijevo pod name upišemo ime našeg programa, a desno odaberemo C project. Pritisak na tipku OK nas vodi u dijaloški okvir u kojem odabiremo gdje ćemo pohraniti naš program. Pa spremite ga negdje... Nakon Save, otvara nam se slijedeće sučelje:



Opala... ovaj je već nešto zapisao! Moja osobna preporuka vam je da ovo `int argc, char *argv[]` što se nalazi unutar zagrada main-a (označeno plavim na slijedećoj slici) odmah pobrišete:

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    system("PAUSE");
    return 0;
}
```

Pa ćemo to i učiniti:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    system("PAUSE");
    return 0;
}
```

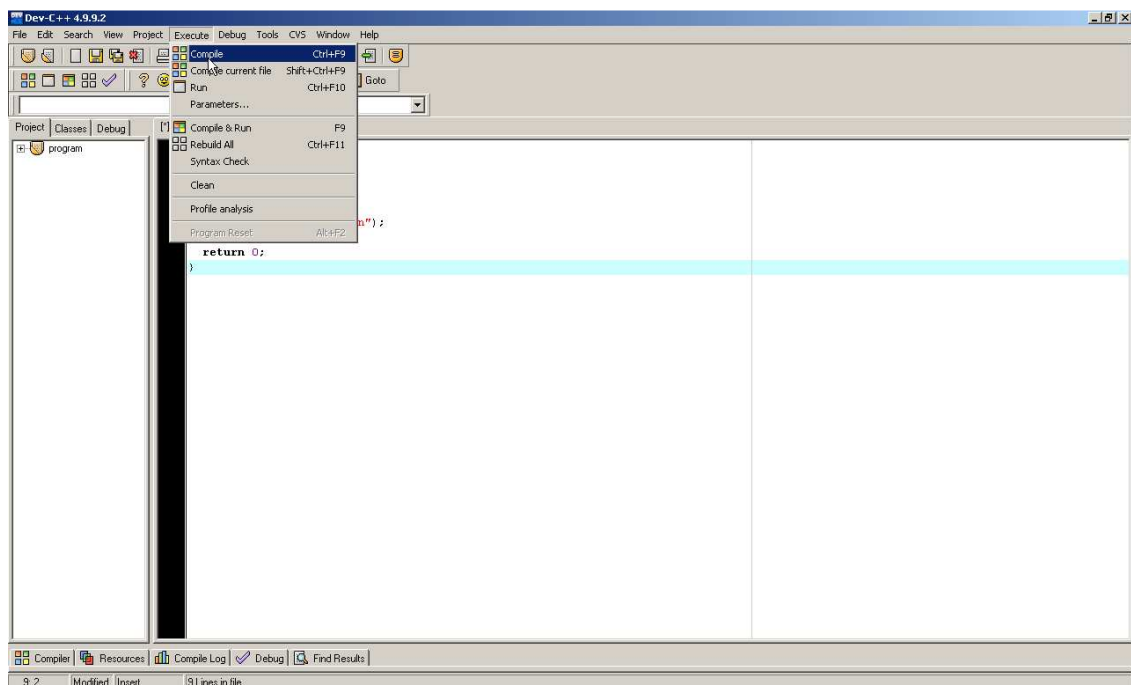
E sad, dakle sam Dev C++ je malo čudna biljka. Naime Kao što vidimo on je sam dodao neke stvari, a najupadljivije je ovo `system („PAUSE“)`; To je naredba koja će u biti zaustaviti program prije njegova završetka. Naime Dev C++ neće po završetku programa ostaviti onaj crni okvir i ispisati poruku „Press any key to continue“ nego će izvršiti, prikazati taj ekran i odmah ga vrlo brzo zatvoriti. Ako kojim slučajem niste Chuck Norris ili Marko Čupić, od rezultata vašeg rada nećete vidjeti ništa. Stoga je tu

taj System („PAUSE“); koji će zaustaviti izvođenje. Sam program napišemo prije te naredbe. U našem slučaju to izgleda ovako:

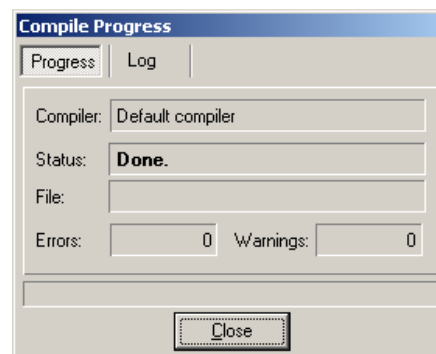
```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf ("Hello world!\n");
    system("PAUSE");
    return 0;
}
```

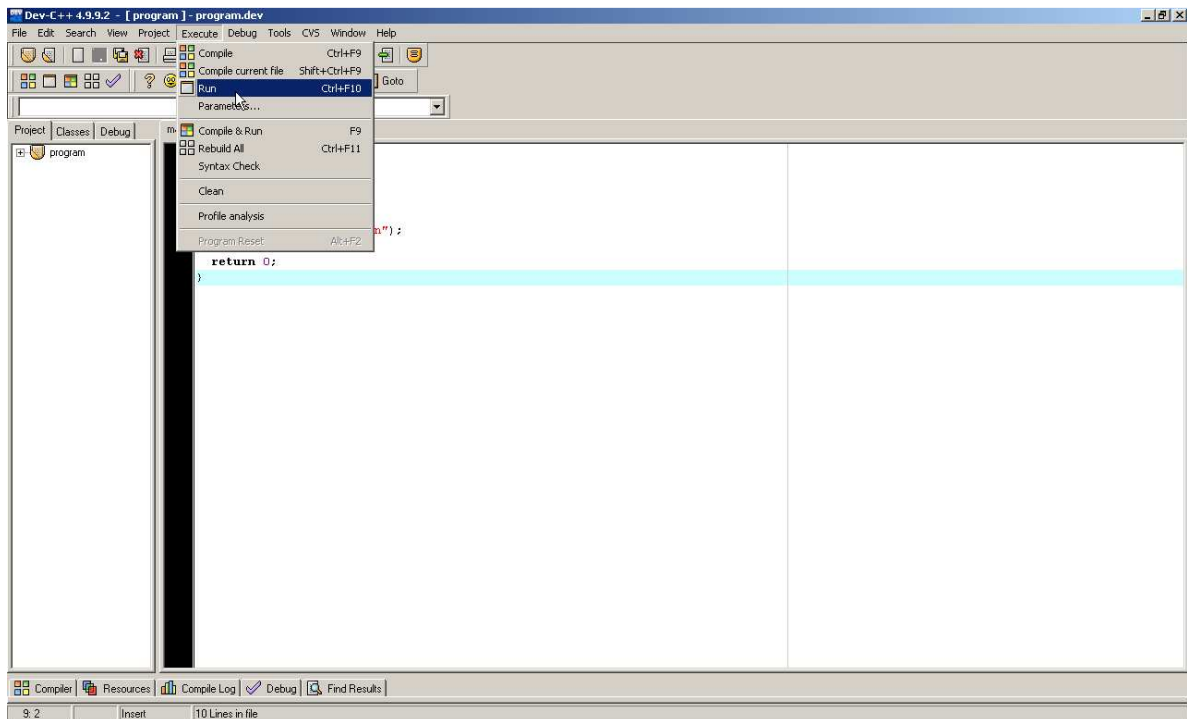
Ne morate se previše zamarati ni ovim stdlib.h na početku, jednostavno, Dev C++ očito voli neke stvari napraviti umjesto nas. Ovo je kod koji sada možemo pokrenuti. To činimo tako da iz alatne trake odaberemo Execute izbornik i u njemu odaberemo opciju Compile.



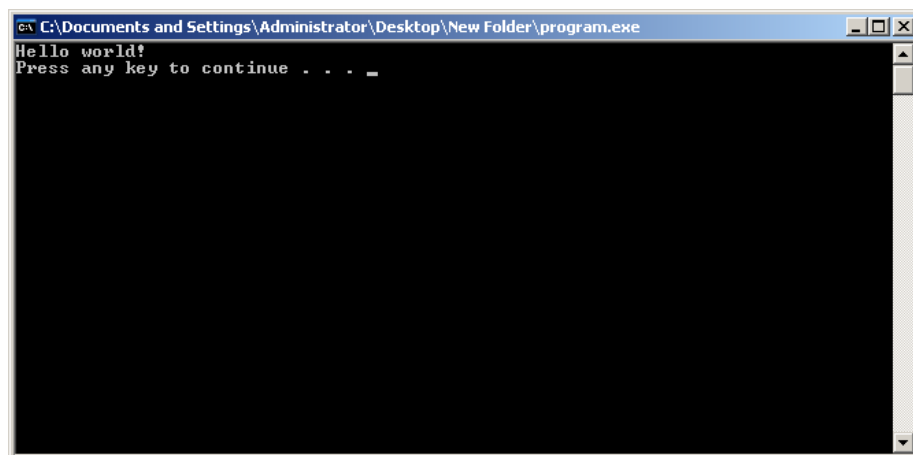
Potom će nam ponuditi da ponovo spremimo, pa vi spremite sam izvršni kod. Nakon toga otvara se okvir u kojem on ispisi neke informacije. Po završetku i uspješnom kompajliranju, taj okvir izgleda ovako:



Status „Done.“ je jako dobra vijest jer to znači da nije bilo pogrešaka. Sad idemo pokrenuti program. To činimo tako da opet odaberemo izbornik Execute i tamo odaberemo opciju „Run.“



Otvora nam se naš poznati prozor:



Idemo sada isprobati ono što sam maloprije pričao. Zatvorite vaše remek-djelo i obrišite onaj redak system („PAUSE“); tako da vaš kod izgleda ovako:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf ("Hello world!\n");
    return 0;
}
```

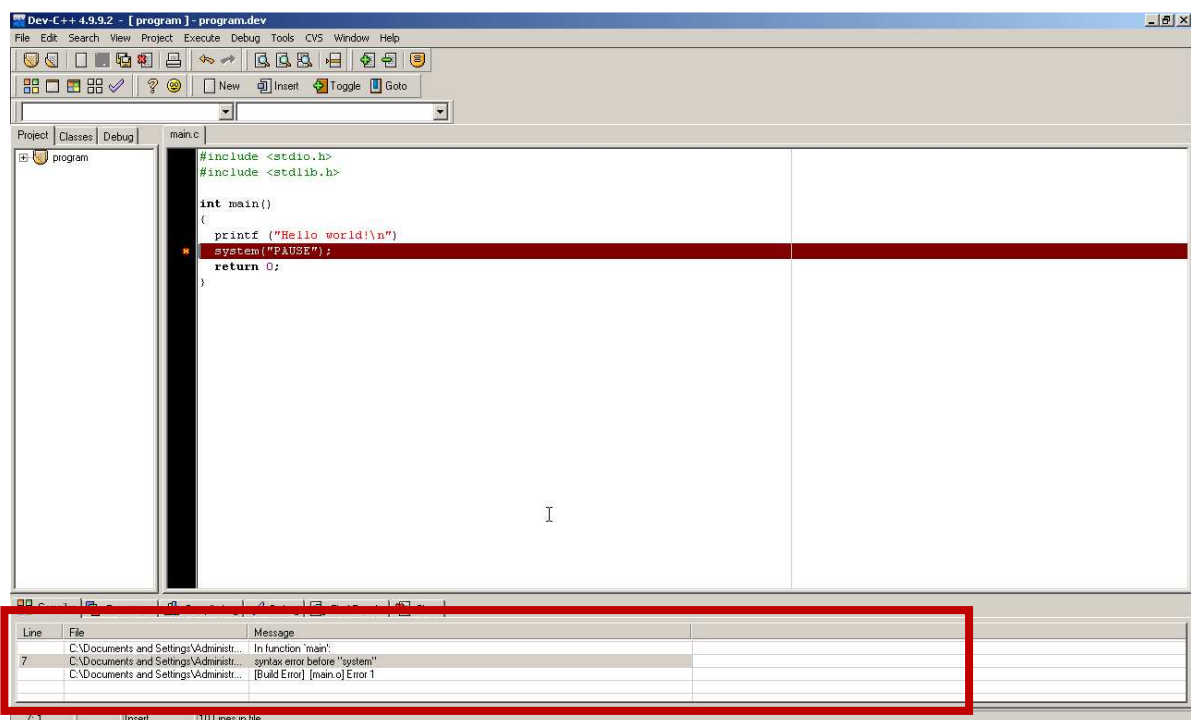
Potom ponovite postupak. Execute → Compile i potom Execute - > Run. Program se pokrenuo i potom odmah zatvorio. Jeste li vidjeli što pročitati? Vjerujem da ne. Zato je važno imati taj system („PAUSE“); u kodu kada radimo u Dev C++. Stoga ga vratite, možete i sa Undo.

Sada ćemo još pogledati kako izgledaju sintaktičke greške u Dev C++. Obrišimo ponovo točka zarez „;“ nakon našeg printf-a u kojem ispisujemo „Hello world!“

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf ("Hello world!\n")
    system("PAUSE");
    return 0;
}
```

Pokrenimo program, odnosno prije pokretanja moramo ga kompajlirati. Stoga odaberimo izbornik Execute i tamo odaberemo već poznatu opciju Compile. Pokreće se kompajliranje, međutim vrlo brzo se onaj okvir zatvara, i vraća nam se naše sučelje u kojem je jedan redak crven. To zorno pokazuje slika:



Vidimo da nam je crven redak poslije printf-a. Pogledajmo što piše u uokvirenom dijelu:

Line	File	Message
7	C:\Documents and Settings\Administr...	In function 'main':
	C:\Documents and Settings\Administr...	syntax error before "system"
	C:\Documents and Settings\Administr...	[Build Error] [main.o] Error 1

Kaže syntax error before „system“. Ovdje nam neće pružiti informaciju kao u slučaju Visual Studia gdje nam je dotični javio da nam nedostaje baš „;“, ali mi ćemo kao vrhunski programeri odmah prepoznati grešnicu koja se ispriječila izvođenju našeg programa. Stoga dodajmo „;“ na kraj retka, kompajlirajmo i pokrenimo program.

Vratimo se još za trenutak na system („PAUSE“); i recimo jedno bitno upozorenje. Kada ćete predavati zadaće u ahyco, onda svakako obrišite ovu liniju koda! Dakle napišite kod, isprobajte ga i kada se uvjerite da radi, maknite taj reda i kopirajte sve u ahyco. I uživajte u vašim bodovima!

4. KOJI KORISTITI?

Odgovor na ovo pitanje morate dati ponajprije sami. Visual studio je vrlo profesionalan alat i mi ćemo na ovom predmetu koristiti tek 1% mogućnosti. Međutim vrlo je težak za vaš tvrdi disk (oko 2 GB) kao i za radnu memoriju vašeg računala prilikom rada u programu. S druge strane Dev C++ je vrlo lagan (tek oko 20 MB), a jednako tako lagan i za memoriju vašeg računala. Ujedno je jednostavniji za korištenje, a može sve što može i Visual Studio (barem ono što nama treba). Mana je eto ta linija koda koju morate dodavati ukoliko želite vidjeti što ste uopće napravili. Druga mana je što se ovaj proizvod koji je imao potencijala kako sada stvari stoje – više ne razvija. Naravno osim ovih alata postoje još mnogi drugi koji su možda i bolji, ali i lošiji, međutim ova dva su najpopularnija kad su studenti PIPI-ja u pitanju.