

ЛАБОРАТОРИЙН АЖИЛ

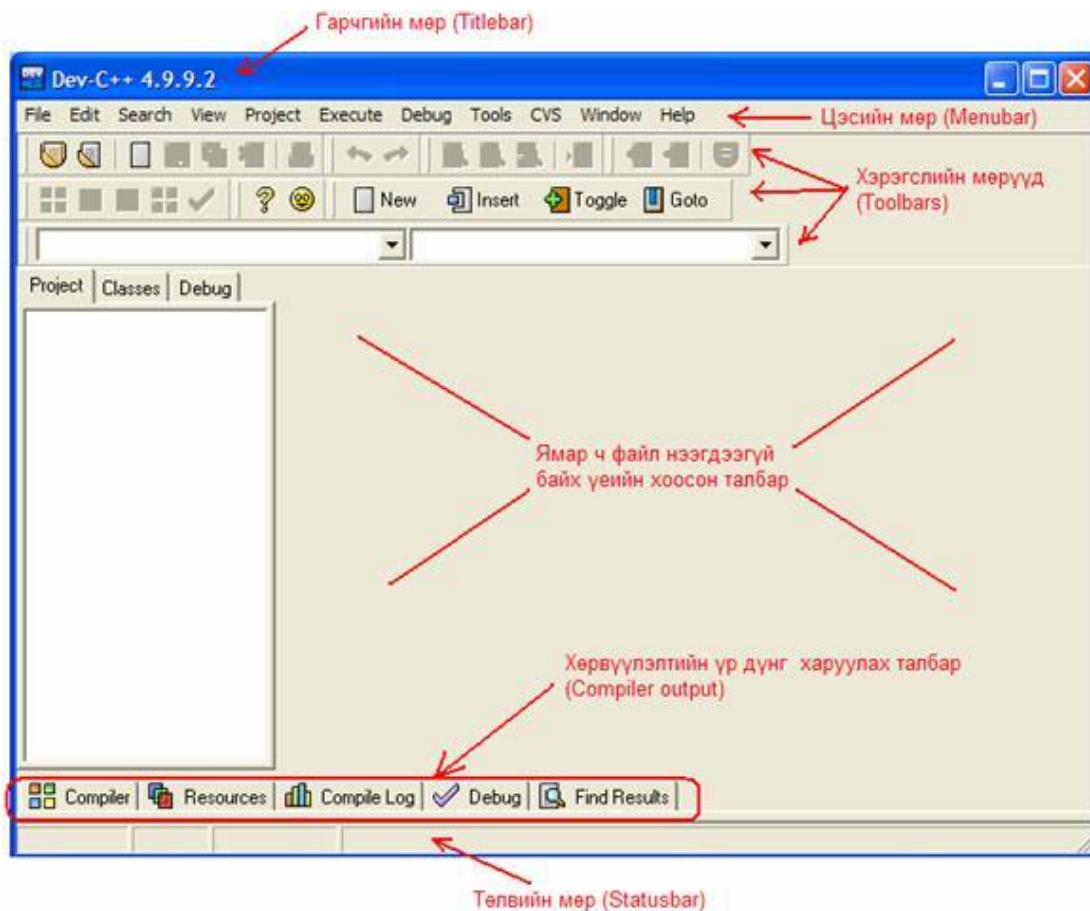
Сэдэв: Dev C++ (Дев Си плас плас) орчинтой Товч танилцах, printf функцийг ашиглан дэлгэцэнд тэмдэгт хэвлэх

Сэдвүүд:

1. [Шинэ файл үүсгэх](#)
2. [Код бичих](#)
3. [Файл хадгалах](#)
4. [Хөрвүүлэх ба Холбох \(Compile and Link\)](#)
5. [“Холбох” гэсэн нэр томъёоны талаар](#)
6. [Ажиллуулах \(Run\)](#)
7. [Хөрвүүлэлтийн алдаа \(Error\)](#)
8. [Хөрвүүлэлтийн сануулга \(Warning\)](#)



Dev C++ нь C болон C++ хэл дээр програм бичихэд зориулагдсан, интернетэд (www.bloodshed.net) үнэгүй тараагддаг програмчлалын систем юм.



Энэ цонх бол бидний ажиллах орчин юм. Цонхны бүтцийг харцгаая (дээрх зураг). Windows орчны ихэнх програмын адилаар цонхны дээд талд гарчгийн мөр, үндсэн цэс, хэрэгслийн мөрүүд байрласан байгаа.

Програмын эх кодыг бичиж, хөрвүүлж, ажиллуулахад шаардлагатай бүх л үйлдлүүд цэс ба хэрэгслийн мөрөнд байрлана.

Шинэ файл үүсгэх

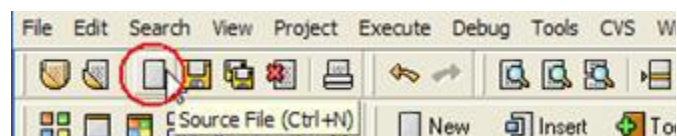
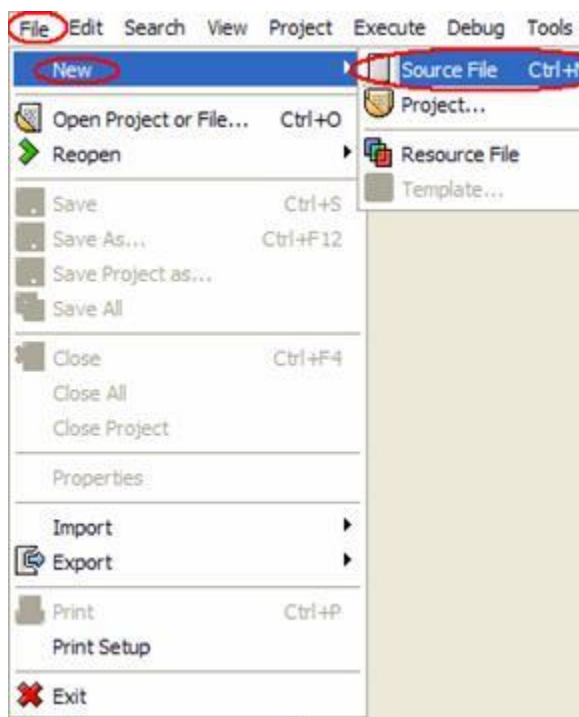
Бид програмын кодыг файл дотор бичих ёстой. Үүний тулд шинээр файл үүсгэнэ. Яагаад заавал файл дотор бичих ёстой гэж?

Бид програмынхаа эх кодыг байнга хадгалж байхын тулд, хүссэн үедээ ашиглахын тулд компьютерын тогтмол санах ойд хадгалах хэрэгтэй биз дээ. Гэтэл аливаа мэдээлэл тогтмол санах ойд яаж хадгалагддаг билээ? Файл хэлбэрээр хадгалагддаг.

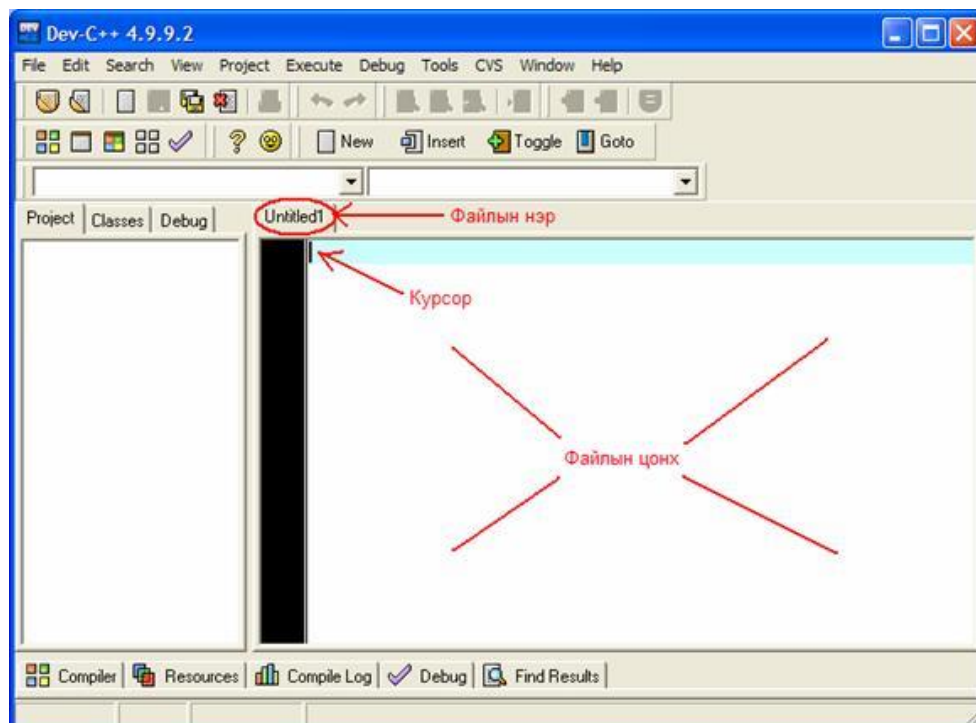
Си-програмын эх кодыг агуулагч файлыг эх файл (Source file) гэдэг. Эх файл нь **c** эсвэл **cpp** гэсэн өргөтгөлтэй байна.

Dev C++ орчинд, шинэ эх файл үүсгэхдээ:

- ☐ File|New|Source File гэсэн үйлдлийг хийнэ,
- ☐ эсвэл хэрэгслийн мөрөөс харгалзах хэрэгслийг сонгоно,
- ☐ эсвэл гарын CTRL+N товчны хослолыг дарна:

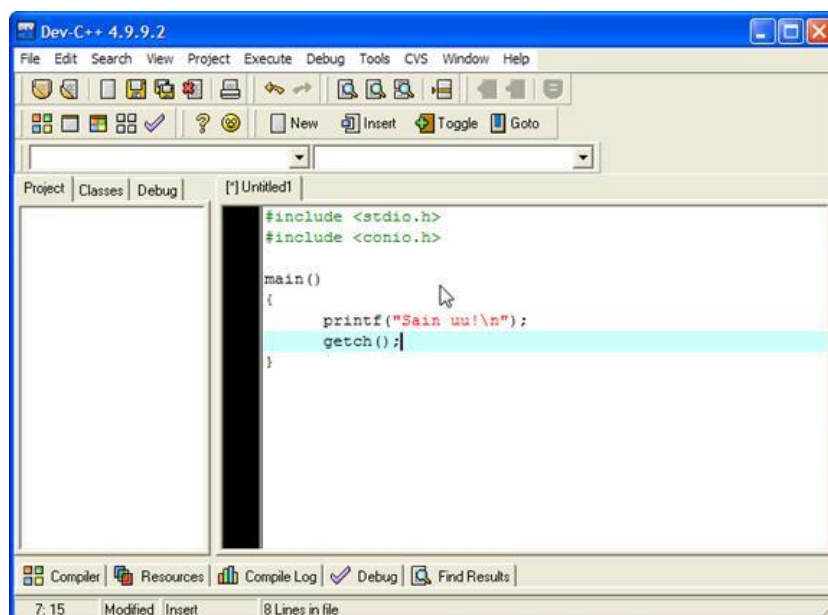


Шинээр үүссэн эх файл автоматаар Untitled гэсэн нэрийг авна. Хэрэв системийг ачаалснаас хойш анх удаа шинэ файл үүсгэж буй бол 1 гэсэн дугаар (Untitled1), хоёр дахь удаагаа шинэ файл үүсгэсэн бол 2 (Untitled2) гэсэн дугаартай г.м.-ээр үргэлжилдэг. Дараах зурагт, анх удаа шинээр үүссэн файлын цонхыг харуулав:



Код бичих

Одоо үүсгэсэн файл дотроо “Сайн уу” хэмээх жишээ програмыг бичье:



Си хэлний хөрвүүлэгч том, жижиг үсгийг ялгаатай гэж үздэг учраас энд текстийг дан жижгээр бичсэн байна. Түүнчлэн текстийн зарим үгнүүд, тухайлбал препроцессорын директив ногоон өнгөөр, тэмдэгт мөр улаан өнгөөр ялгагджээ.

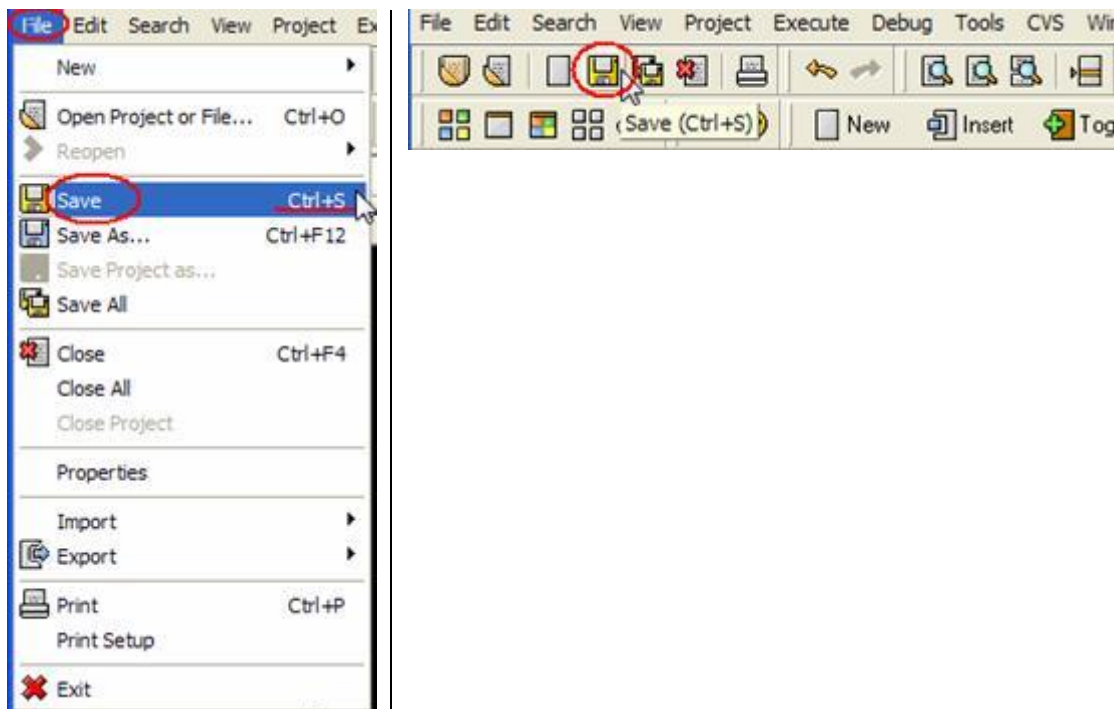
Програмын кодыг бичиж байхдаа текстийг нааш цааш хуулах, зөөх шаардлагууд гарч болно. Тийм тохиолдолд Edit цэсний Cut (CTRL+X), Copy (CTRL+C), Paste (CTRL+V) г.м. командуудыг ашиглах нь тохиромжтой.

Хамгийн сүүлд хийсэн засвараа буцааж, хүчингүй болгоё гэвэл мөн Edit цэсний Undo (CTRL+Z) командыг ашиглах хэрэгтэй.

Файл хадгалах

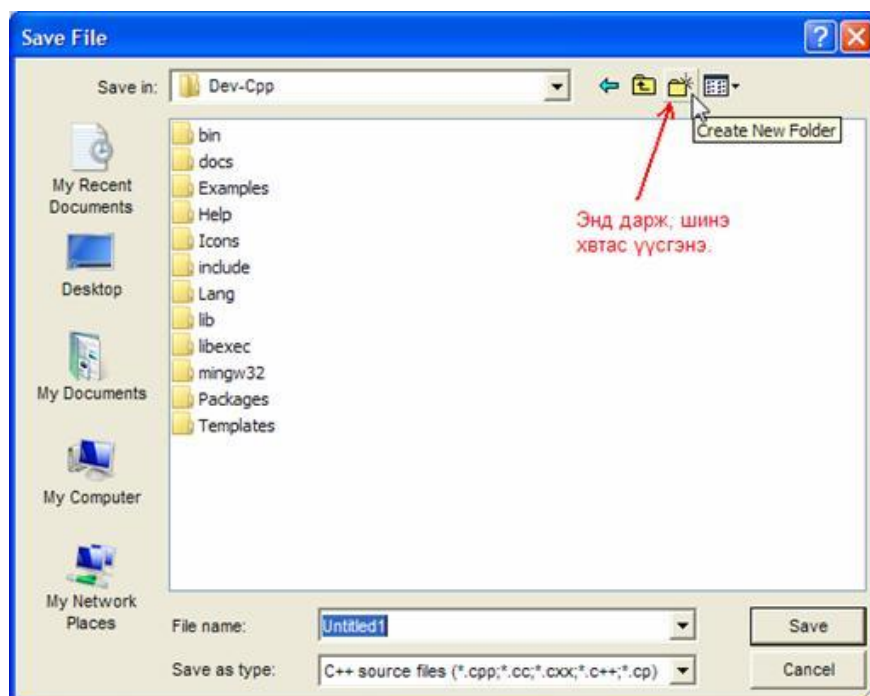
Текстээ бичиж дуусаад файлаа хадгалах хэрэгтэй. Үүний тулд:

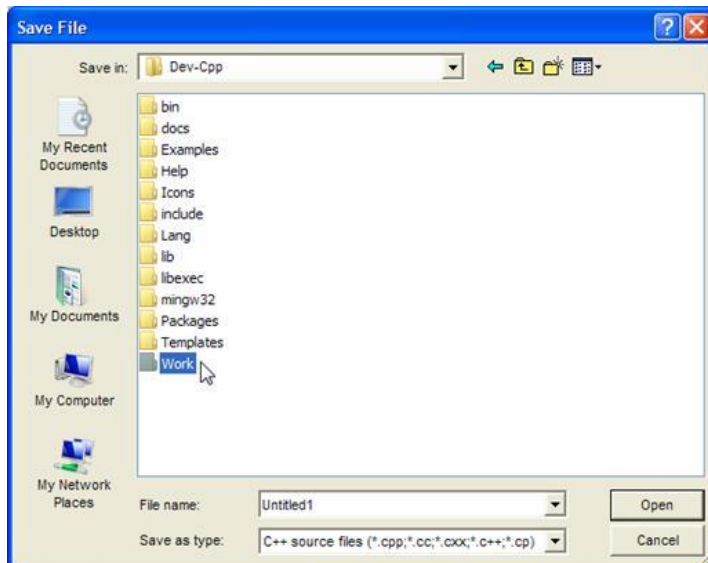
- ☐ File|Save гэсэн үйлдлийг хийнэ,
- ☐ эсвэл хэрэгслийн мөрөөс харгалзах хэрэгслийг сонгоно,
- ☐ эсвэл гарын CTRL+S товчны хослолыг дарна:



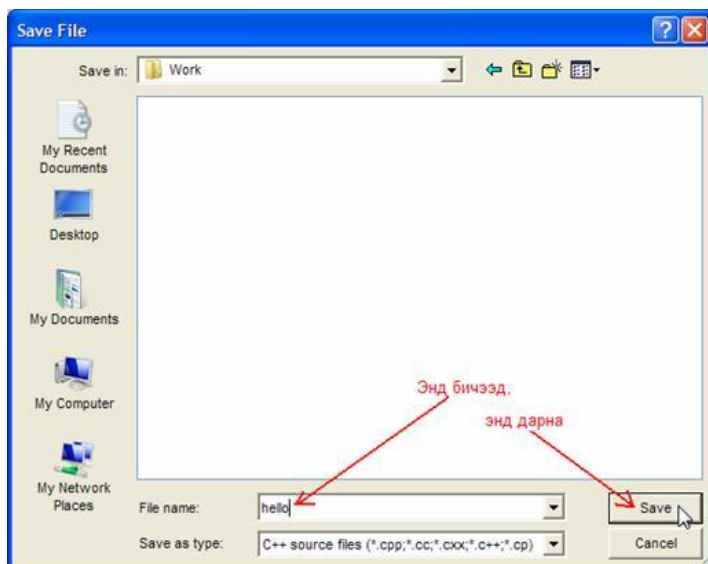
Ингэхэд Save File гэсэн цонх гарч ирнэ. Энэ цонхны File Name талбарт хадгалах нэрээ бичиж өгөөд Save товчийг дардаг.

Файлууд автоматаар C:\Dev-Cpp хавтсанд хадгалагддаг. Харин бид Dev-Cpp хавтсанд дахиад нэг Work гэсэн нэртэй шинэ хавтас үүсгэж, түүн дотор файлаа хадгалъя. Шинэ хавтас хэрхэн үүсгэхийг дараах зурагт үзүүлэв:





Одоо үүсгэсэн хавтас руугаа нэвтэрч орох хэрэгтэй. Тэгээд файлынхаа нэрийг бичиж хадгална. Бид файлаа **hello** гэж нэрлэе. Харин өргөтгөлийг бичих шаардлагагүй. Систем өөрөө автоматаар **cpp** гэсэн өргөтгөлийг нэрийн ардаас залгадаг:

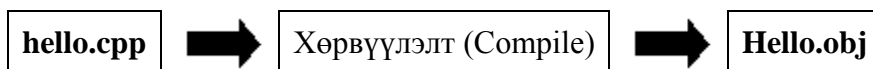


Ингэснээр жишээ програм маань C:\Dev-Cpp\Work гэсэн замаар hello.cpp гэсэн файл болон хадгалагдаж байгаа юм.

Хөрвүүлэх ба Холбох (Compile and Link)

Одоо бид бичсэн эх кодоо хөрвүүлж объект код гаргаж авах хэрэгтэй. Текст маань эх файл дотор агуулагдаж байгаа учраас эх файлаа хөрвүүлнэ гэсэн үг. Хөрвүүлэхэд объект файл

(object file) үүсдэг. Дотроо объект кодыг агуулж байдаг файлыг объект файл гэнэ. Си-программын объект файл нь **obj** гэсэн өргөтгөлтэй байдаг. Харин нэр нь хэвээрээ байна. Хөрвүүлэх процессыг схемчлэн үзүүлбэл:

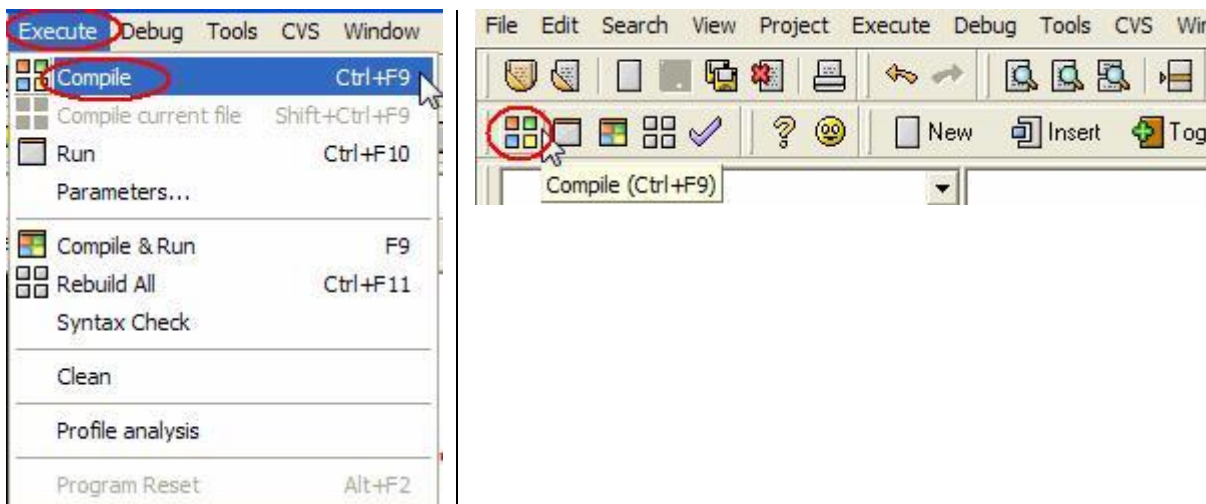


Дараа нь, үүсгэсэн объект файлаасаа ажиллах модуль (execution module) буюу биелэх програмыг гаргаж авах ёстой. Энэ бол манай эцсийн “бүтээгдэхүүн” бөгөөд **exe** гэсэн өргөтгөлтэй, адил нэртэй файл байдаг. Exe файл бол бидний анх бичсэн програмын 100% машины хэл дээрх хувилбар юм. Програмчлалын нэр томъёогоор, объект модулиас ажиллах модуль үүсгэх энэ алхмыг холболт хийх (Link) гэж нэрийддэг. Холболтыг холбогч (linker) хэмээх програм гүйцэтгэдэг. Үүнийг схемчлэн үзүүлбэл:



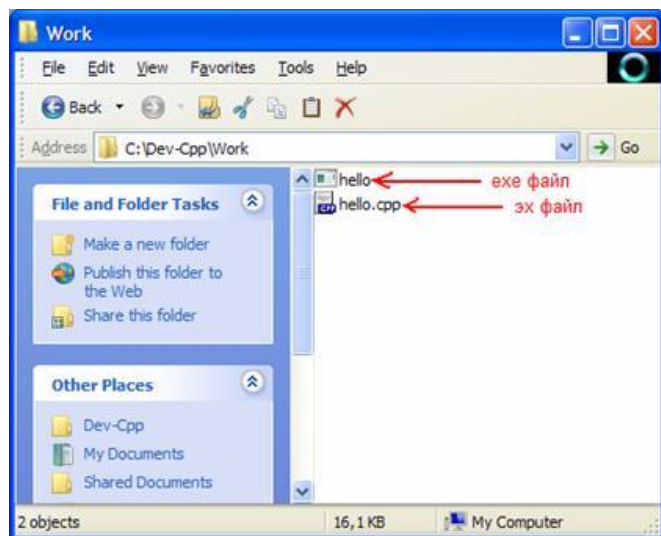
Тэгвэл Dev C++ орчинд Хөрвүүлэлт ба Холболтыг хэрхэн гүйцэтгэх вэ?

- ☐ Execute|Compile гэсэн үйлдлийг хийнэ,
- ☐ эсвэл хэрэгслийн самбараас харгалзах хэрэгслийг сонгоно,
- ☐ эсвэл гарын CTRL+F9 товчны хослолыг дарах хэрэгтэй:



Ингэхэд систем өөрөө эхлээд Хөрвүүлэгчийг дуудаж, хэрэв хөрвүүлэлт амжилттай болвол Холбогчийг дуудаж ажиллуулаад, амжилттай болсон тохиолдолд **hello.exe** файлыг үүсгэдэг байна.

Ө.х. завсрын бүтээгдэхүүн болох объект файлыг үүсгэлгүй шууд “эцсийн бүтээгдэхүүн”-ийг гаргадаг ажээ. **Ехе** файл хатуу дискэнд, эх файл байрлах тэр хавтсанд үүснэ. Манай тохиолдолд энэ нь C:\Dev-Cpp\Work хавтас юм:



“Холбох” гэсэн нэр томъёоны талаар

Яагаад “Холбох” гэсэн нэр томъёо хэрэглэгдэж байна вэ? Яагаад гэвэл энэ алхам нь объект файлд Си хэлний стандарт сангийн функцуудыг холбож өгдөг учраас тэр. Гэхдээ препроцессорын үйлдлээс ялгаатай нь, зөвхөн програмд ашиглагдаж буй тэр функцуудын код объект файлд холбогдоно (нийлнэ). Тиймээс одоо бол эх кодоос биелэх програм гаргаж авах хүртэлх бүх алхмыг дараах байдлаар нэгтгэн схемчилж болно:



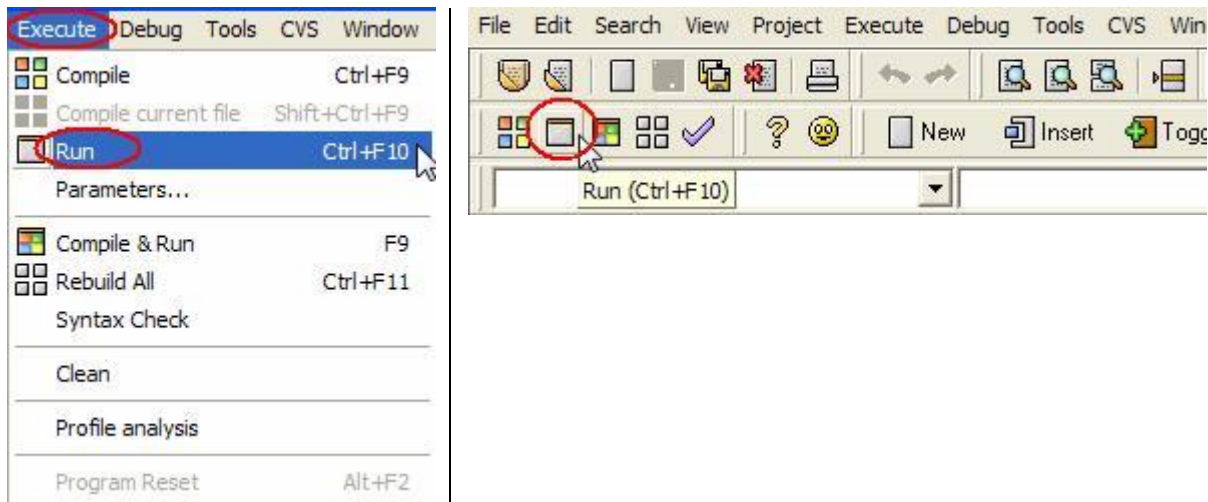
Ажиллуулах (Run)

Эцэст нь бид үүсгэсэн **exe** файлаа ажиллуулах ёстой. Ингэснээр бидний бичсэн програмыг компьютер биелүүлж эхлэх юм. Үүний тулд Work хавтас руу орж, **hello.exe** файлыг сонгон ачаалж болно.

Харин Dev C++ орчин бидэнд системээс гаралгүйгээр шууд exe файлаа дуудаж ажиллуулах боломжийг олгодог. Үүний тулд:

- ☐ Execute|Run үйлдлийг хийх,
- ☐ эсвэл хэрэгслийн мөрөөс хэрэгсэл сонгох,
- ☐ эсвэл гарын CTRL+F10 товчны хослолыг дарах

хэрэгтэй:



Тэгэхэд програм ажиллаж эхлэнэ.

Програм биелж эхлэхэд дэлгэцэнд юу гарахыг доорх зурагт үзүүлэв:

```
Sain uu!  
_
```

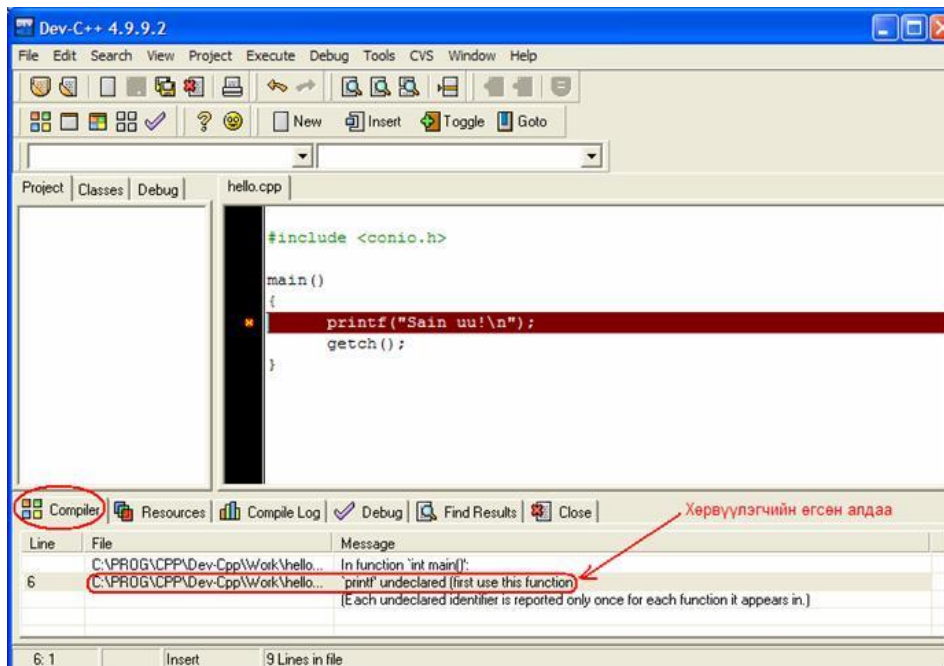
. Дэлгэцийг бүхэлд нь эзэлсэн хар дэвсгэртэй цонх гарч ирэх бөгөөд хамгийн дээд талын мөрөнд Sain uu! гэсэн мэндчилгээ хэвлэгджээ. Ингээд програм “пауз” авсан байна. ENTER (эсвэл өөр нэг товч) дарвал програм төгсөж, буцаад Dev C++ орчны цонх гарч ирэх болно.

Хөрвүүлэлтийн алдаа (Error)

Ямар нэг шалтгаанаар эх кодонд дүрмийн алдаанууд (syntax errors) агуулагдаж байна гэе. Ийм тохиолдолд хөрвүүлэлт амжилттай болдоггүй. Ж.нь дараах зургийг харцгаая.



Одоо энэ кодыг хөрвүүлэх гэж үзье. Тэгвэл хөрвүүлэгч ажилласны дараа цонх дараах байдалтай болно:



Юу болсон байна вэ?

Файлын цонхонд, програмын текст доторх алдаатай мөрийг ялгаж харуулсан байна:



Файлын цонхны доод талд байрлах Compiler output талбарын Compiler самбар идэвхжин, хөрвүүлэгчийн өгч буй алдааны мессежийг хэвлэсэн байна. Энэ нь:

“hello.cpp файлын 6-р мөрөнд үл танигдах функц ашигласан байна”

гэсэн алдаа байна.

Учир нь бид **printf()** функцыг ашигласан боловч түүний тухай мэдээллийг агуулагч толгой файлыг програмд оруулж ирээгүй байна билээ. Тиймээс хөрвүүлэгч энэ функцыг таньж чадахгүй байгаа юм.

Ингээд засвараа хийж (`#include <stdio.h>` гэсэн директивийг бичээд) дахин хөрвүүлбэл “бүх зүйл Окей” болно. Хөрвүүлэгчийн өөр нэгэн чухал үүрэг эндээс харагдаж байна. Түүний тусламжтайгаар текстэнд агуулагдах аливаа алдааг илрүүлдэг юм. Ялангуяа том хэмжээний текстэнд энэ нь онцгой ач холбогдолтой.

Хөрвүүлэлтийн сануулга (Warning)

Заримдаа хөрвүүлэгч нь алдаа өгөхгүй боловч сануулга өгөх тохиолдлууд байж болно. Сануулга ба алдааны ялгаа нь, хэдийгээр сануулга өгөгдсөн ч бүтээгдэхүүн-файл үүсдэг.

Тэглээ гээд хөрвүүлэгчийн өгч буй сануулгуудыг тоолгүй орхиж болохгүй. Нэг сануулгаас болж програм буруу ажиллаад эхлэж магадгүй юм. Тиймээс гараад байгаа сануулгуудын шалтгааныг олж засах хэрэгтэй. Програм зохиогчийн гол зорилго бол 0 сануулга, 0 алдаатай програм бичих явдал билээ.

СЭДЭВ: PRINTF ФУНКЦИЙГ АШИГЛАН ДЭЛГЭЦЭНД ТЭМДЭГТ ХЭВЛЭХ

Зорилго: Програмчлалын Си хэлний дэлгэцэнд мэдээлэл гаргах printf функцыг ашиглан дэлгэцэнд тэмдэг гаргаж туршилт хийх.

Онолын хэсэг:

printf функцийн бичигдэх ерөнхий хэлбэр: `int printf(char
*format, arg1, arg2 . . .);`

format - тухайн өгөгдөл ямар хэлбэрээр дэлгэцэнд гарахыг тодорхойлдог. format тэмдэгт мөр нь шууд дэлгэцэнд хэвлэгдэх ердийн тэмдэгтүүд болон хэвшүүлэлтийн тэмдэгтүүдээс тогтоно.

arg1 - өгөгдөл 1 **arg2** -
өгөгдөл 2

буцаах утга - printf функц нь амжилттай хэвлэсэн тэмдэгтийн тоог буцаадаг.

Доорх хvснэгтэнд курсорыг удирдах тэмдэгдvүдийг харуулав.

Тэмдэгт	Тайлбар
\n	Курсорыг шинэ мөр
\b	Курсорыг 1 тэмдэгтээр буцаах
\t	Tab тэмдэгт (Хэд хэдэн хоосон зай)
\r	Курсорыг мөрийн эхэнд
\a	Дуут дохио гаргах
\\	\ тэмдэгтийг өөрийг нь хэвлэх
\"	" тэмдэгт хэвлэх

Жишээ дасгал

Жишээ 1:

Бодолт:

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

```
int main()
```

```
{
```

```
    printf("12345678\n");
```

```
// шинэ мөр авах
```

```
    printf("12345 78\b\b\b6\n");
```

```
// Хоосон зайн оронд 6-г хэвлэх
```

```
    printf("12345\t678\n");
```

```
// Тоонуудын хооронд зай авах
```

```
    printf("12345\r678\n");
```

```
// 123 гэсэн тоог дарж бичнэ
```

```
    printf("\\Hi\\ \n");
```

```
    printf("\\\"Hi\" \n");
```

```
    return 0;
```

```
}
```

Програмын үр дүн:

```
12345678
```

```
12345678
```

12345 678
67845 \Hi\ "Hi"

Жишээ 2: Од (*) тэмдэгтүүд ашиглан өндөр нь 7 тэмдэгт, өргөн нь 5 тэмдэгт бүхий E блокыг хэвлэх програм бич

```
*****
*
*
*****
*
*
*****
```

Бодолт:

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

Бие дааж хийх ажлууд

Дасгал 1: "HELLO" блокыг хэвлэх програм зохио; нэг үсэг нь 7 тэмдэгтийн өндөр(7-н мөр), 5 тэмдэгтийн өргөнтэй байна.

Дасгал 2: Дараах зургийг хэвлэх програм бич.

