

C/C++ programlarını Visual Studio Code (VS Code) üzerinde çalıştırmak ve derlemek için birkaç adım izlenmesi gerekmektedir.

#### A. Windows'ta C/C++ derleme ortamını kurmak için gerekli adımlar:

### 1. MinGW (GCC) Derleyicisini Yükle

C ve C++ programlarını derlemek için MinGW derleyicisini kurulması gerekmektedir. MinGW, GCC'nin Windows için bir uyarlamasıdır.

#### **MinGW'yi yüklemek için:**

1. **MinGW İndir:** "[MinGW Installation Manager](#)"ı indir.
2. **Kurulum:** İndirilen dosyayı çalıştır ve "Basic Setup" altında şunları seç:
  - mingw32-base
  - mingw32-gcc-g++
3. **Seçilen bu iki paketi indir ve kur.**
4. **Yolu Ayarla:**
  - "C:\MinGW\bin" dizinini sistem yoluna (PATH) eklemelisin:
    - **Başlat Menüsü > Sistem Özellikleri > Gelişmiş Sistem Ayarları > Ortam Değişkenleri** bölümüne git.
    - "Sistem Değişkenleri" altında **Path**'i bul ve düzenleye tıkla.
    - Yeni bir satır ekle ve "C:\MinGW\bin" yaz.
    - Bilgisayarını yeniden başlat.

### 2. VS Code'u Yükle

Eğer Visual Studio Code yüklü değilse [buradan](#) indirip kurabilirsin.

### 3. C/C++ Eklentisini Yükle

VS Code üzerinde C/C++ geliştirme yapabilmek için C/C++ uzantısını eklemelisin.

1. VS Code'u aç.
2. **Extensions** (Eklentiler) bölümüne git (sol taraftaki blok simgesi).
3. Arama kutusuna "C/C++" yaz ve **C/C++ Microsoft eklentisini** kur.

#### 4. VS Code'da Build and Run İçin Gerekli Ayarları Yap

VS Code'un bir terminal üzerinden C ve C++ programlarını derleyip çalıştırması için tasks.json ve launch.json dosyalarını yapılandırman gerekiyor.

##### 1. tasks.json Dosyası (Build Komutu)

- Proje dizininde .vscode adlı bir klasör oluştur ve içerisine tasks.json dosyası ekle.
- İçerisine şu kodu yaz:

```
{
  "version": "2.0.0",
  "tasks": [
    {
      "label": "C/C++: gcc.exe build active file",
      "type": "shell",
      "command": "gcc",
      "args": [
        "-g",
        "${file}",
        "-o",
        "${fileDirname}\\${fileBasenameNoExtension}.exe"
      ],
      "group": {
        "kind": "build",
        "isDefault": true
      },
      "problemMatcher": ["$gcc"],
      "detail": "Generated task by Debugger."
    }
  ]
}
```

Bu yapılandırma, geçerli dosyayı derlemek için gcc komutunu kullanır.

## 2. launch.json Dosyası (Debug Komutu)

- Proje dizininde .vscode klasörünün içerisine launch.json dosyasını ekle.
- İçerisine şu kodu yaz:

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "(gdb) Launch",
      "type": "cppdbg",
      "request": "launch",
      "program": "${fileDirname}\\${fileBasenameNoExtension}.exe",
      "args": [],
      "stopAtEntry": false,
      "cwd": "${fileDirname}",
      "environment": [],
      "externalConsole": true,
      "MIMode": "gdb",
      "setupCommands": [
        {
          "description": "Enable pretty-printing for gdb",
          "text": "-enable-pretty-printing",
          "ignoreFailures": true
        }
      ],
      "preLaunchTask": "C/C++: gcc.exe build active file",
      "miDebuggerPath": "C:\\\\MinGW\\\\bin\\\\gdb.exe",
      "internalConsoleOptions": "neverOpen",
      "logging": { "engineLogging": true }
    }
  ]
}
```

Bu dosya, gdb hata ayıklayıcıyı kullanarak programın nasıl çalıştırılacağını tanımlar.

## 5. Terminal Üzerinden C veya C++ Kodunu Derleme ve Çalıştırma

1. Yeni bir C dosyası oluştur, örneğin hello.c:

```
#include <stdio.h>

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

2. **Terminal** > **Yeni Terminal** menüsünden terminal aç.
3. Dosyanı derlemek için şu komutu kullanabilirsin:

```
gcc hello.c -o hello.exe
```

4. Çalıştırmak için:

```
./hello.exe
```

## 6. VS Code Üzerinden Build ve Run

- **F5** tuşuna basarak programını derleyebilir ve çalıştırabilirsin.
- **Terminalde** veya **Debug** bölümünde derleme ve çalışma hatalarını izleyebilirsin.

Bu adımları takip ederek Windows üzerinde VS Code ile C/C++ programları derlenip çalıştırılabilir.

## B. Mac'te C/C++ derleme ortamını kurmak için gerekli adımlar:

### 1. Xcode Command Line Tools Yükle

Mac üzerinde GCC yerine genellikle **Clang** kullanılır ve bu araçlar Xcode Command Line Tools ile birlikte gelir. Xcode Command Line Tools, Apple'ın sunduğu ücretsiz bir derleme araçları setidir.

**Xcode Command Line Tools'u yüklemek için:**

1. **Terminali aç.**
2. Aşağıdaki komutu çalıştırarak Xcode Command Line Tools'u yükle:

```
xcode-select --install
```

3. Ekranı bir yükleme penceresi gelecektir, "Install" butonuna tıklayarak yüklemeyi tamamla.

Bu adım, Clang ve diğer temel derleyici araçlarını yükler.

### 2. VS Code'u Yükle

Eğer Visual Studio Code yüklü değilse, [VS Code](#) sitesinden indirip kurabilirsiniz.

### 3. C/C++ Eklentisini Yükle

VS Code üzerinde C/C++ geliştirme yapmak için Microsoft'un sunduğu C/C++ uzantısını eklemen gerekiyor.

1. **VS Code'u aç.**
2. Sol taraftaki **Extensions (Eklentiler)** simgesine tıkla (blok simgesi).
3. Arama kutusuna **C/C++** yaz ve **C/C++ Microsoft eklentisini** kur.

### 4. VS Code Üzerinde C/C++ Derleyip Çalıştırmak İçin Gerekli Ayarları Yap

VS Code'u kullanarak C/C++ programlarını derlemek ve çalıştırmak için tasks.json ve launch.json dosyalarını ayarlamamız gerekiyor.

#### 1. tasks.json Dosyasını Oluştur (Build İşlemi İçin)

- **VS Code'da** proje dizininde .vscode adlı bir klasör oluştur ve içerisine tasks.json adlı bir dosya ekle.
- Bu dosyaya aşağıdaki kodu ekle:

```
{
  "version": "2.0.0",
  "tasks": [
    {
      "label": "C/C++: clang build active file",
      "type": "shell",
      "command": "/usr/bin/clang",
      "args": [
        "-g",
        "${file}",
        "-o",
        "${fileDirname}/${fileBasenameNoExtension}"
      ],
      "group": {
        "kind": "build",
        "isDefault": true
      },
      "problemMatcher": ["$gcc"],
      "detail": "Generated task by Debugger."
    }
  ]
}
```

Bu, mevcut açık dosyayı Clang ile derlemek için bir görev tanımlar.

## 2. launch.json Dosyasını Oluştur (Debug İşlemi İçin)

- Proje dizinindeki .vscode klasörünün içine launch.json dosyasını oluştur.
- Bu dosyaya aşağıdaki kodu ekle:

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "(lldb) Launch",
      "type": "cppdbg",
      "request": "launch",
      "program": "${fileDirname}/${fileBasenameNoExtension}",
      "args": [],
      "stopAtEntry": false,
      "cwd": "${fileDirname}",
      "environment": [],
      "externalConsole": true,
      "MIMode": "lldb",
      "preLaunchTask": "C/C++: clang build active file",
      "miDebuggerPath": "/usr/bin/lldb",
      "setupCommands": [
        {
          "description": "Enable pretty-printing for lldb",
          "text": "type format add -f pretty --summary-string\n\"\\\" --category C++ STL",
          "ignoreFailures": true
        }
      ],
      "internalConsoleOptions": "openOnSessionStart"
    }
  ]
}
```

Bu dosya, LLDB hata ayıklayıcıyı kullanarak programın nasıl çalıştırılacağını tanımlar. LLDB, Xcode Command Line Tools ile birlikte gelen hata ayıklayıcıdır.

## 5. Terminalde C veya C++ Kodunu Derleyip Çalıştırma

1. Bir .c veya .cpp dosyası oluştur, örneğin hello.c:

```
#include <stdio.h>

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

2. Terminalde bu dosyayı Clang ile derleyebilirsiniz:

```
clang hello.c -o hello
```

3. Çalıştırmak için:

```
./hello
```

## 6. VS Code Üzerinden Build ve Run

- **F5** tuşuna basarak VS Code üzerinde programını derleyip çalıştırabilirsin.
- Eğer hata ayıklamak istersen, terminalde hata mesajlarını görebilirsin ve Debug bölümü üzerinden LLDB kullanarak hata ayıklayabilirsin.

### Ekstra Adımlar:

- Derleme işlemlerini kolaylaştırmak için VS Code'un sağladığı "debugger" özelliğini kullanabilir ve kodunuzu interaktif bir şekilde hata ayıklayabilirsin.