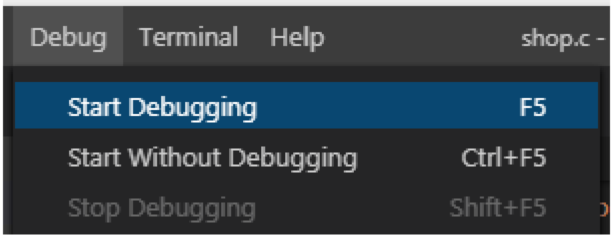
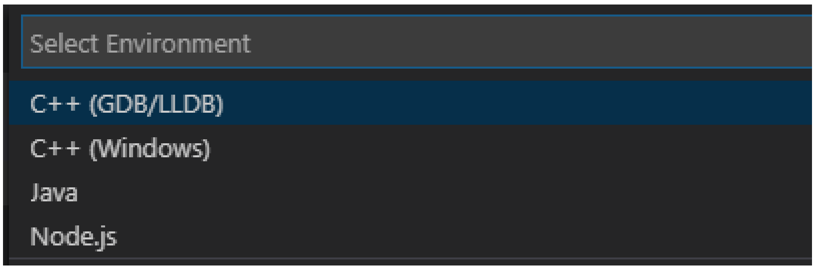
# GCC Debugger setup in VS Code

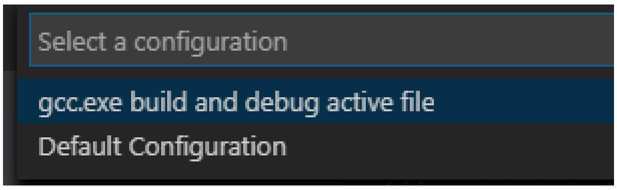
1. Open C file in VS Code
2. Set breakpoint in code
3. Press F5 or select Start debugging from Debug menu



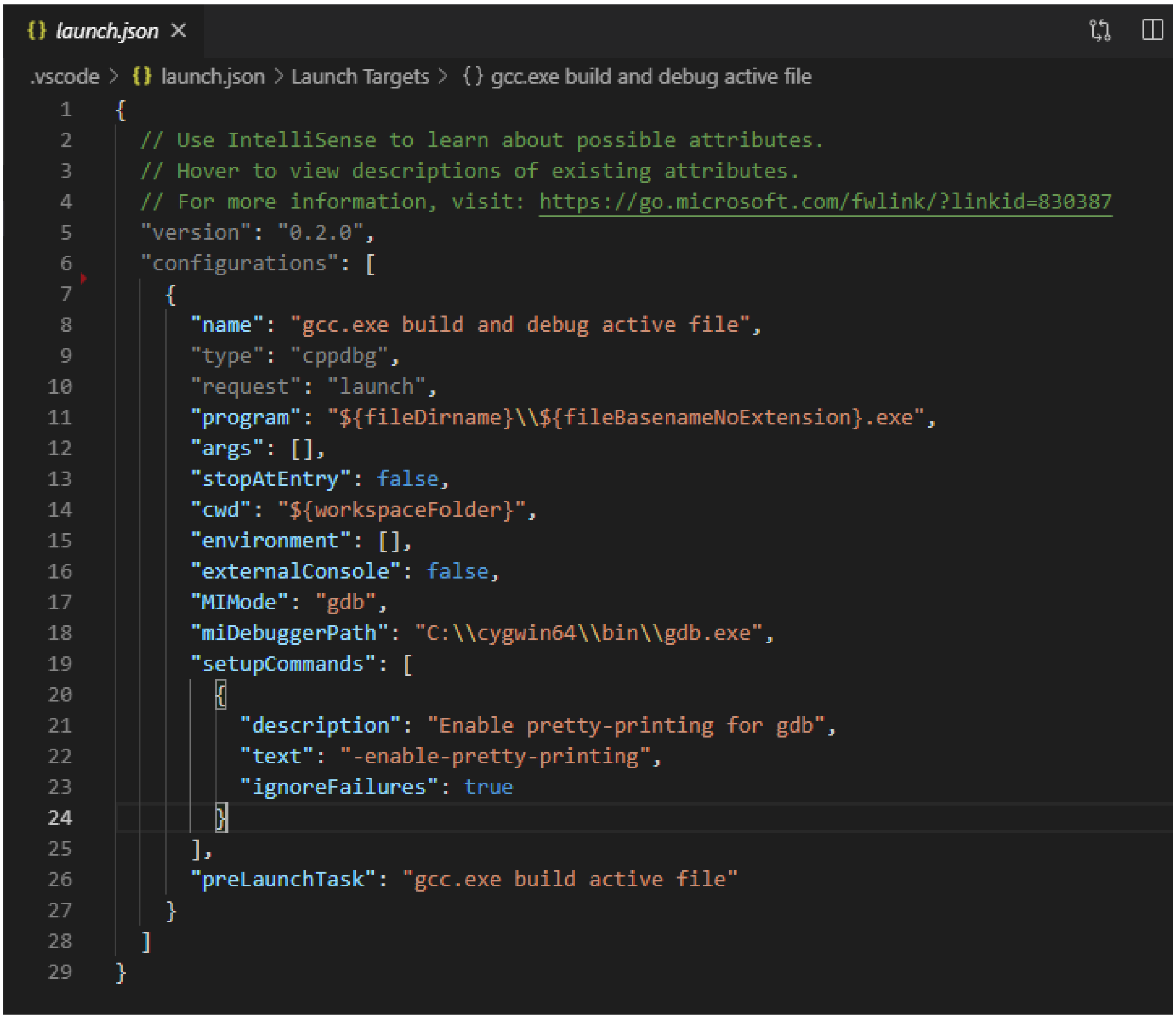
4. Select C++(GDB/LLDB) from popup



5. Then select gcc.exe build and debug active file

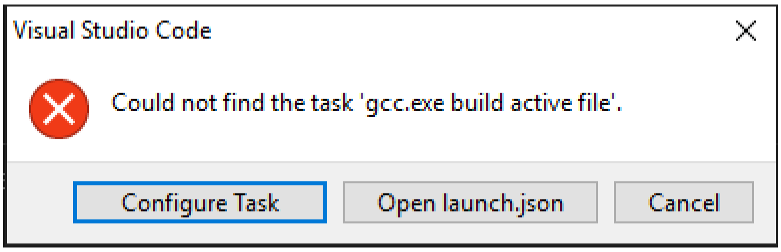


6. The Launch.json window should now pop up, see that this has the right path to your gdb.exe file, save and close.


{} launch.json ×
.vscode > {} launch.json > Launch Targets > {} gcc.exe build and debug active file

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "name": "gcc.exe build and debug active file",
            "type": "cppdbg",
            "request": "launch",
            "program": "${fileDirname}\\${fileBasenameNoExtension}.exe",
            "args": [],
            "stopAtEntry": false,
            "cwd": "${workspaceFolder}",
            "environment": [],
            "externalConsole": false,
            "MIMode": "gdb",
            "miDebuggerPath": "C:\\cygwin64\\bin\\gdb.exe",
            "setupCommands": [
                {
                    "description": "Enable pretty-printing for gdb",
                    "text": "-enable-pretty-printing",
                    "ignoreFailures": true
                }
            ],
            "preLaunchTask": "gcc.exe build active file"
        }
    ]
}
```

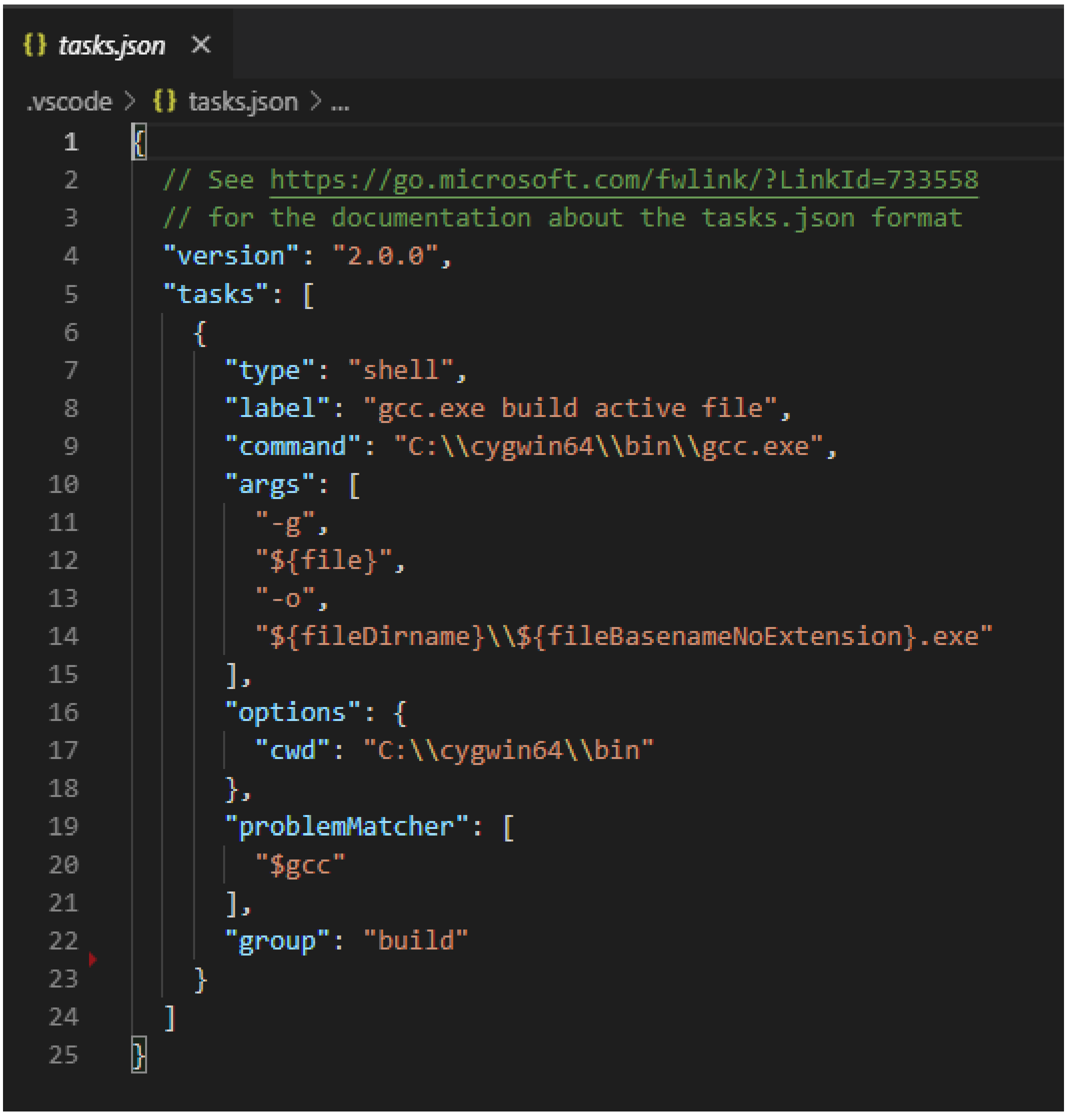
7. G back to the C file and F5 again, ow it should pop up a message about not being able to build the active file, click “Configure Task”.



Could not find the task 'gcc.exe build active file'.

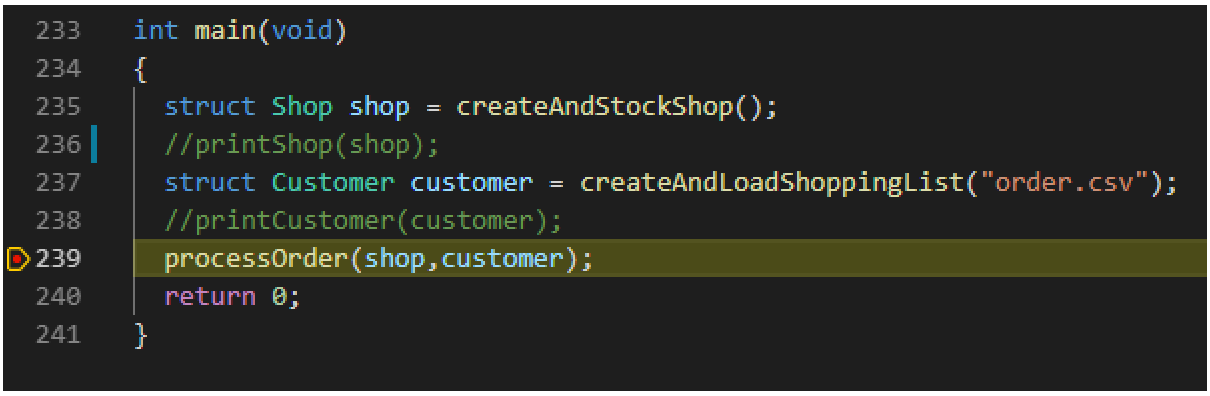
8. Select C/C++:gcc.exe build active file

9. Now the task,json file should pop up and confirm that “command” points to the right location where the gcc.exe file is located. Save and close



```
{
  // See https://go.microsoft.com/fwlink/?LinkId=733558
  // for the documentation about the tasks.json format
  "version": "2.0.0",
  "tasks": [
    {
      "type": "shell",
      "label": "gcc.exe build active file",
      "command": "C:\\cygwin64\\bin\\gcc.exe",
      "args": [
        "-g",
        "${file}",
        "-o",
        "${fileDirname}\\${fileBasenameNoExtension}.exe"
      ],
      "options": {
        "cwd": "C:\\cygwin64\\bin"
      },
      "problemMatcher": [
        "$gcc"
      ],
      "group": "build"
    }
  ]
}
```

10. Go back to the C file again and press F5.
11. Now everything should start up and the code should stop at the debugger breakpoint.



```
int main(void)
{
  struct Shop shop = createAndStockShop();
  //printShop(shop);
  struct Customer customer = createAndLoadShoppingList("order.csv");
  //printCustomer(customer);
  processOrder(shop,customer);
  return 0;
}
```