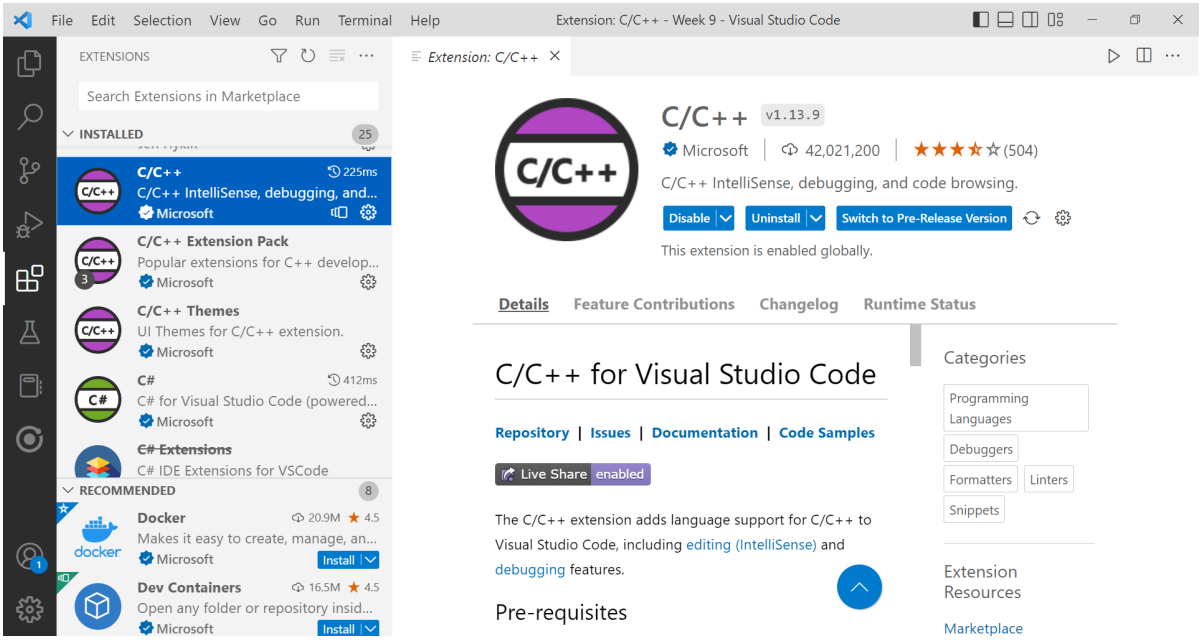
# Programming Fundamentals

Debugger - Week 09

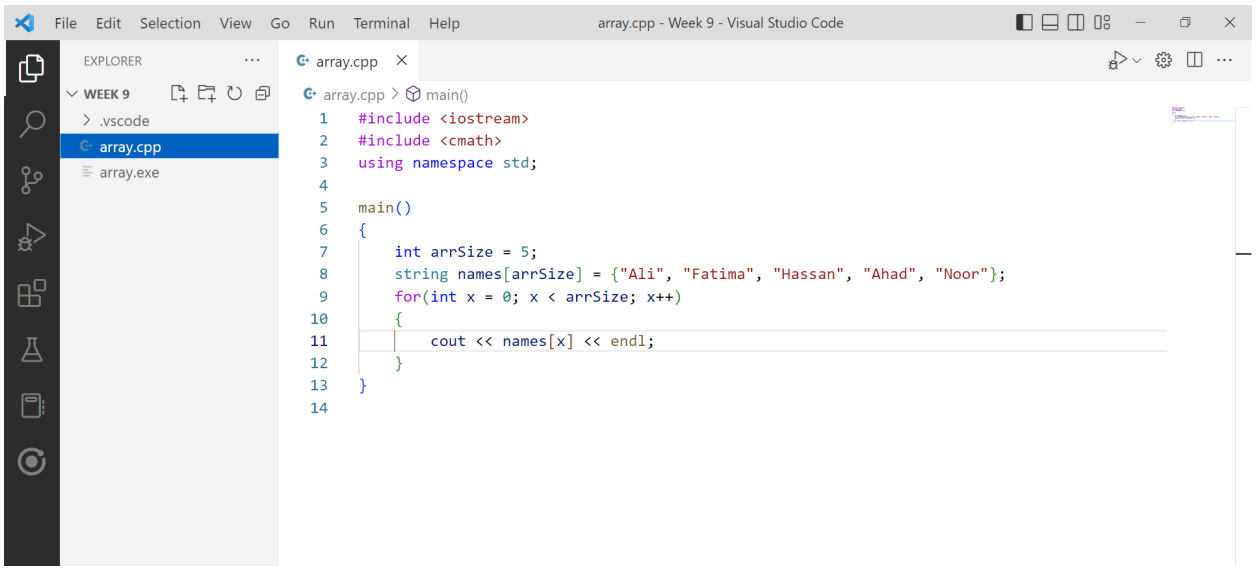
Let’s configure the Debugger in Visual Studio Code for C++.

To configure the debugger in Visual Studio Code for C++, follow these steps:

1. Install the C/C++ extension for Visual Studio Code (All of you have already installed this extension).

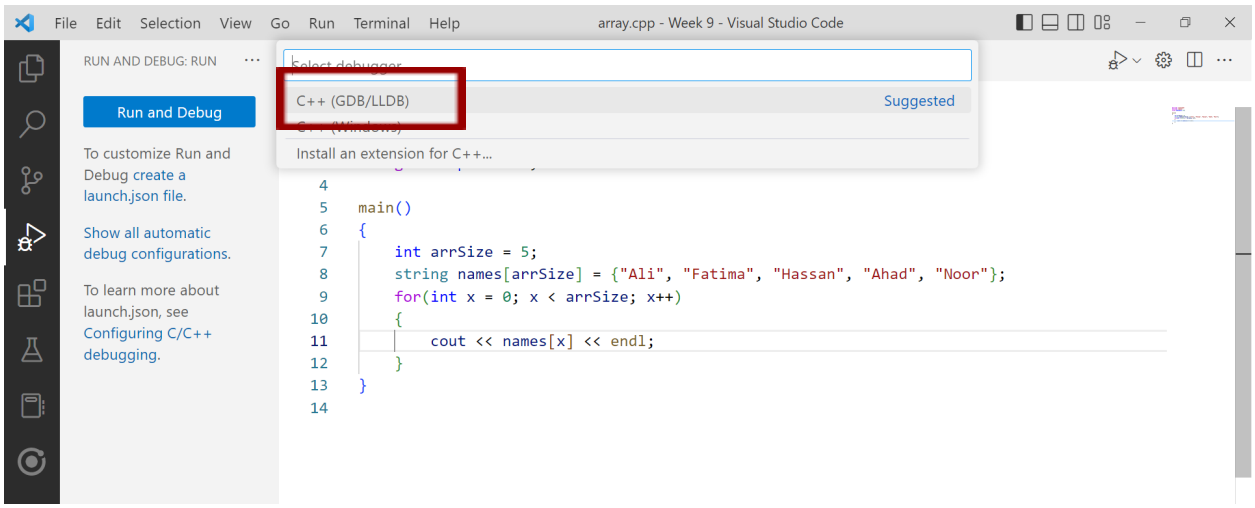



2. Create a new project in Visual Studio Code and create a new file with a .cpp extension.

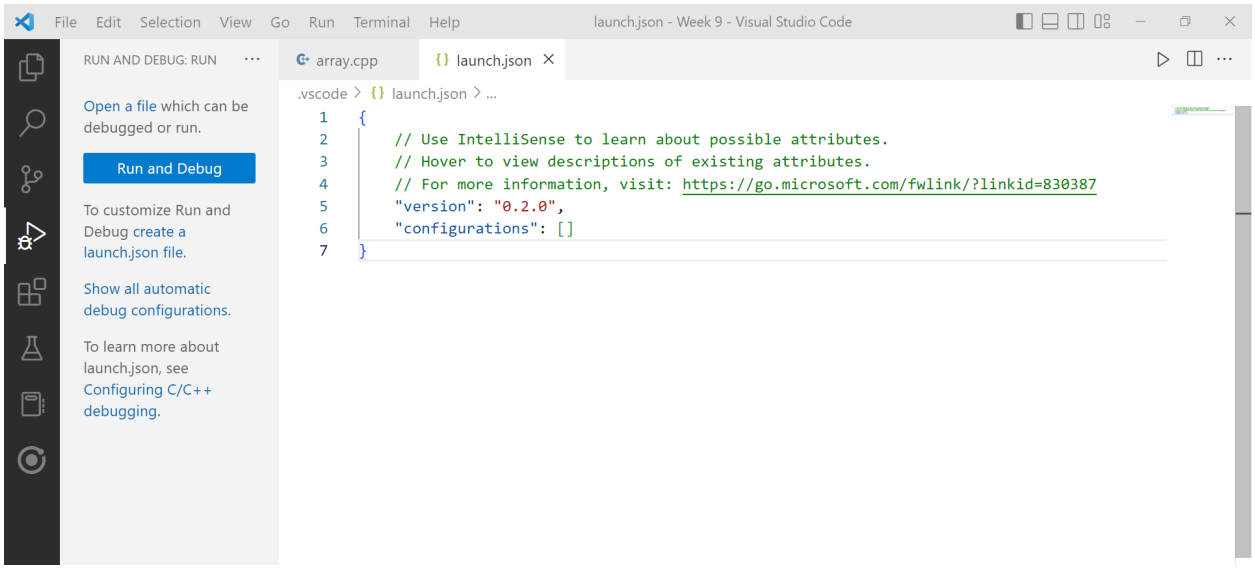



3. Press Ctrl + Shift + D to open the Debug view.


4. Then click the “Create a launch.json File” to create a new launch configuration file (launch.json).




5. Select the C++ (GDB/LLDB) option. A new launch.json file will be opened.



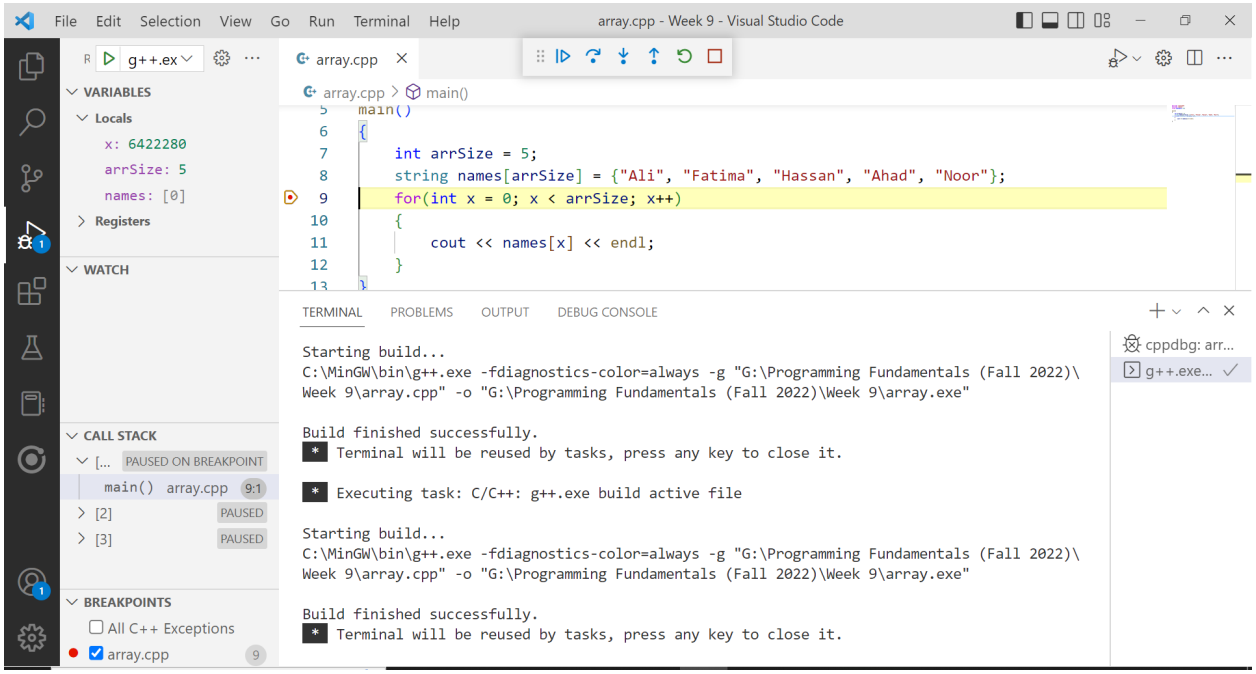

6. In the newly created launch.json file, set the following configuration:

```
{
    // 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": "g++.exe - Build and debug active file",
            "type": "cppdbg",
            "request": "launch",
            "program":
"${fileDirname}\\${fileBasenameNoExtension}.exe",
            "args": [],
            "stopAtEntry": false,
            "cwd": "${workspaceFolder}",
            "environment": [],
            "externalConsole": true,
            "MIMode": "gdb",
            "miDebuggerPath": "C:\\MinGW\\bin\\gdb.exe",
            "setupCommands": [
                {
                    "description": "Enable pretty-printing for
gdb",
                    "text": "-enable-pretty-printing",
                    "ignoreFailures": true
                }
            ],
            "preLaunchTask": "C/C++: g++.exe build active file"
        }
    ]
}
```

7. Press Ctrl + S to save the file.
8. Make sure to replace the path `"miDebuggerPath": "C:\\MinGW\\bin\\gdb.exe"`, with the path to the MinGW bin folder on your system.
9. Now you can set a breakpoint in your code.


10. Press f5 to run the code in the Debug mode.




11. The debugger will stop at the breakpoint, and you can step through your code and inspect variables.

# Programming Fundamentals

Debugger - Week 09


12.  "Step into" - allows you to move into the next function or statement. This will "step into" any function calls within the current line, so that you can see what's happening inside.
13.  "Step out" - allows you to move out of the current function or statement. This will "step out" of the current function call and continue execution at the next line in the parent function.
14.  "Step over" - allows you to skip over the current line, without going into any functions that might be called. This is useful if you don't want to examine the details of a specific function call, but just want to move on to the next line.
15.  "Continue" - allows you to run the program continuously, without stopping for each line, until the next breakpoint is encountered.
16.  "Restart" - restarts the debugging session from the beginning. This is useful if you want to run the debugged process from the start again, for example, after making changes to your code.
17.  "Stop" - stops the current debugging session, killing the process being debugged and terminating the debugger.