

Week 4 – Homework

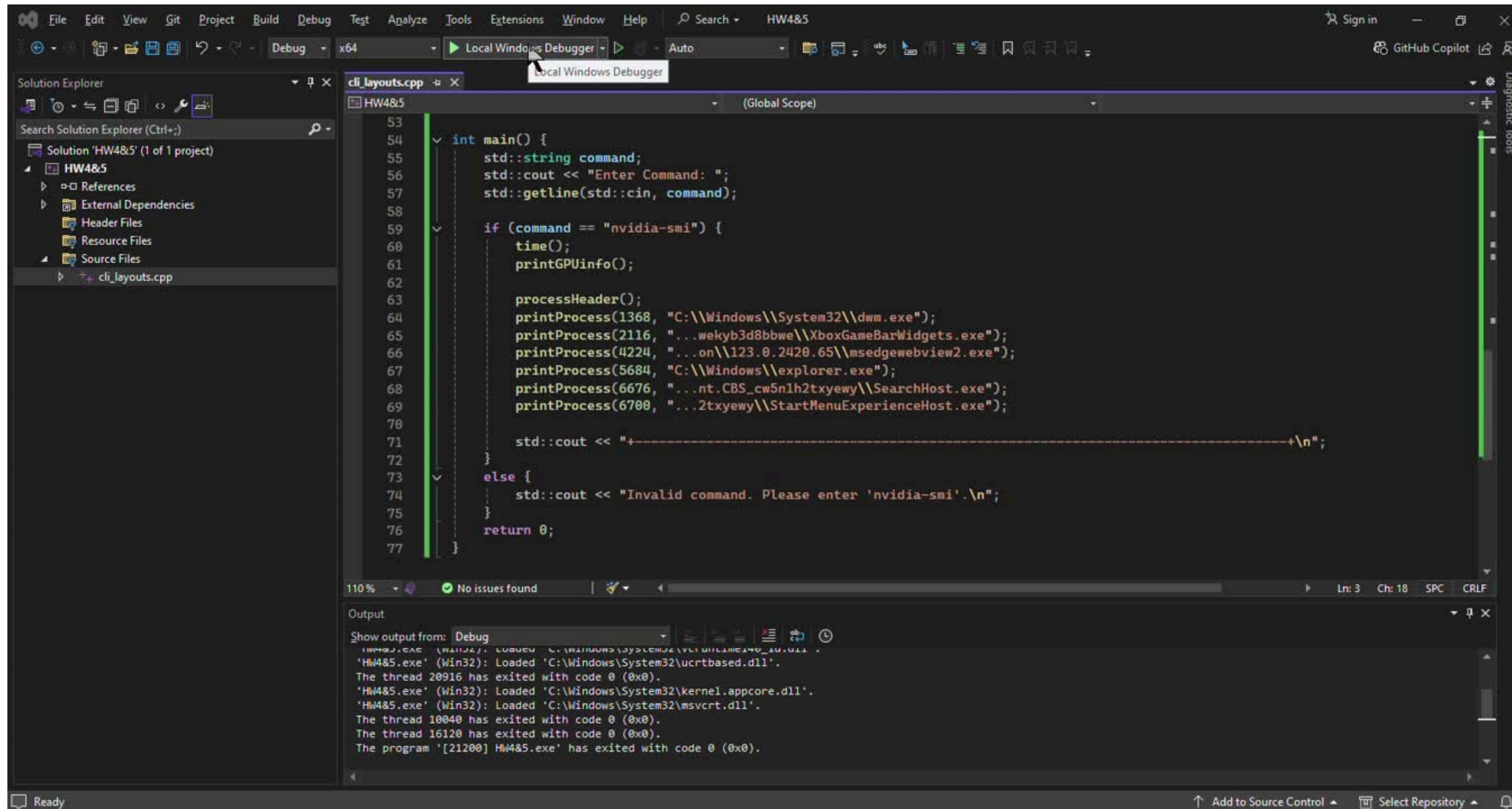
Creating customized CLI layouts

Kai Hiori J. Padilla



video demo

- GOOGLE DRIVE LINK



```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search HW4&5
Debug x64 Local Windows Debugger Auto
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'HW4&5' (1 of 1 project)
  HW4&5
    References
    External Dependencies
    Header Files
    Resource Files
    Source Files
      cli_layouts.cpp
cli_layouts.cpp
53
54 int main() {
55     std::string command;
56     std::cout << "Enter Command: ";
57     std::getline(std::cin, command);
58
59     if (command == "nvidia-smi") {
60         time();
61         printGPUinfo();
62
63         processHeader();
64         printProcess(1368, "C:\\Windows\\System32\\dwm.exe");
65         printProcess(2116, "...wekyb3d8bbwe\\XboxGameBarWidgets.exe");
66         printProcess(4224, "...on\\123.0.2420.65\\msedgewebview2.exe");
67         printProcess(5684, "C:\\Windows\\explorer.exe");
68         printProcess(6676, "...nt.CBS_cw5nlh2txyewy\\SearchHost.exe");
69         printProcess(6700, "...2txyewy\\StartMenuExperienceHost.exe");
70
71         std::cout << "+-----+\n";
72     }
73     else {
74         std::cout << "Invalid command. Please enter 'nvidia-smi'.\n";
75     }
76     return 0;
77 }
110% No issues found Ln: 3 Ch: 18 SPC CRLF
Output
Show output from: Debug
[21200] HW4&5.exe (Win32): Loaded 'C:\\Windows\\System32\\ntdll.dll'.
[21200] HW4&5.exe (Win32): Loaded 'C:\\Windows\\System32\\kernel.appcore.dll'.
[21200] HW4&5.exe (Win32): Loaded 'C:\\Windows\\System32\\ucrtbased.dll'.
[21200] HW4&5.exe (Win32): Loaded 'C:\\Windows\\System32\\kernel.appcore.dll'.
[21200] HW4&5.exe (Win32): Loaded 'C:\\Windows\\System32\\msvcrt.dll'.
[21200] HW4&5.exe (Win32): The thread 20916 has exited with code 0 (0x0).
[21200] HW4&5.exe (Win32): The thread 10040 has exited with code 0 (0x0).
[21200] HW4&5.exe (Win32): The thread 16120 has exited with code 0 (0x0).
[21200] HW4&5.exe (Win32): The program '[21200] HW4&5.exe' has exited with code 0 (0x0).
```

printGPUinfo

- the function's main purpose is to display the GPU info in a specific format and copy the output of the NVIDIA System Management Interface (NVIDIA-SMI) tool.

```
void printGPUinfo() {
    std::cout << "+-----+\n";
    std::cout << "| NVIDIA-SMI 551.86      Driver Version: 551.86      CUDA Version: 12.4      |\n";
    std::cout << "+-----+-----+-----+\n";
    std::cout << "| GPU   Name               TCC/WDDM | Bus-Id        Disp.A | Volatile Uncorr. ECC |\n";
    std::cout << "| Fan  Temp  Perf          Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |\n";
    std::cout << "|                                           | MIG M.         |\n";
    std::cout << "+=====+=====+=====+\n";
    std::cout << "| 0     NVIDIA GeForce GTX 1080  WDDM | 00000000:26:00.0  On |               N/A  |\n";
    std::cout << "| 28%   37C   P8             11W / 180W | 701MiB / 8192MiB |             0%   Default |\n";
    std::cout << "|                                           |               N/A  |\n";
    std::cout << "+-----+\n";
}
```

processHeader

- This functions prints out a header for the processes

```
void processHeader() {  
    std::cout << "+-----+\n";  
    std::cout << "| Processes:                               GPU Memory|\n";  
    std::cout << "| GPU  GI  CI  PID   Type   Process name      Usage      |\n";  
    std::cout << "+-----+\n";  
}
```

printProcess

- This function's main purpose is to format and print the details of a process, ensuring that the output are all aligned and structured properly

```
void printProcess(int pid, std::string process_name) {
    const int maxLineWidth = 79;
    const int fixedPartLength = 29;
    size_t availableSpace = maxLineWidth - fixedPartLength - std::to_string(pid).length();

    std::string formattedName = process_name;

    if (formattedName.length() > availableSpace) {
        formattedName = "..." + formattedName.substr(formattedName.length() - (availableSpace - 3), availableSpace - 3);
    }
    else {
        formattedName += std::string(availableSpace - formattedName.length(), ' ');
    }

    std::cout << "| 0    N/A N/A " << std::setw(4) << pid << "    C+G    " << formattedName << "    N/A    |\n";
}
```

time

- This function just retrieves the current local time

```
void time() {  
    std::time_t t = std::time(nullptr);  
    std::tm local_time;  
    localtime_s(&local_time, &t);  
  
    std::cout << std::put_time(&local_time, "%a %b %d %H:%M:%S %Y") << "\n";  
}
```

End of slides

