Name: Leah Sigana
Assignment 1 & 2

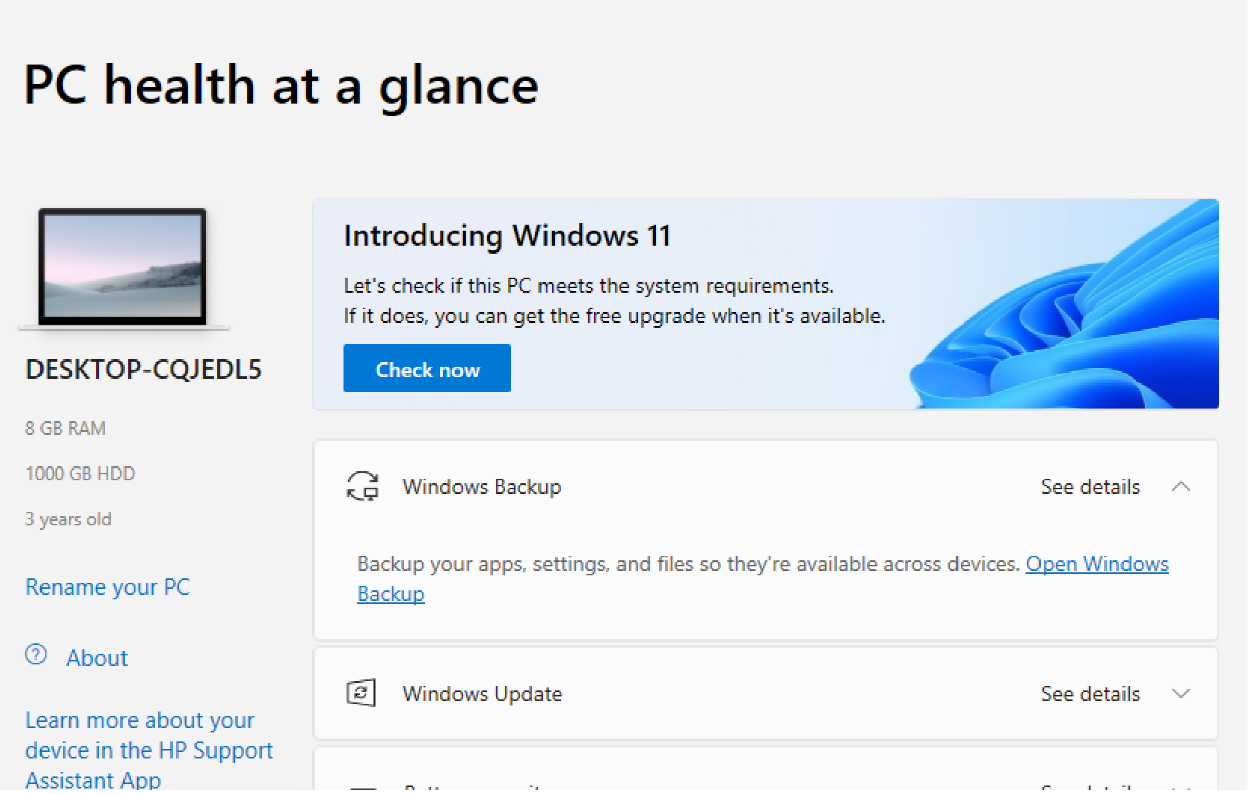
Assignment 1

Tasks:

We'll break down the tasks involved:

1. Selecting Your Operating System (OS):

Choose an OS that aligns with your preferences and project needs. Preferred option windows



PC health at a glance

DESKTOP-CQJEDL5

8 GB RAM

1000 GB HDD

3 years old

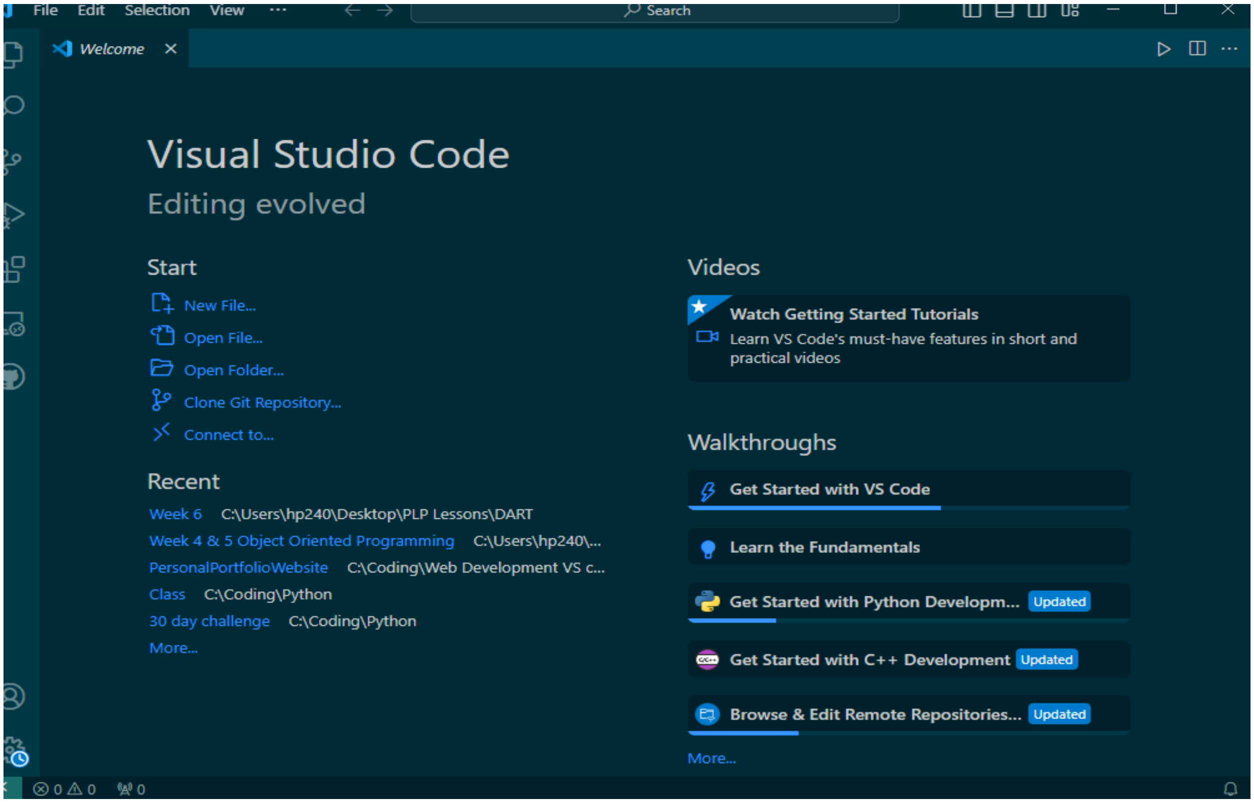
Introducing Windows 11

Let's check if this PC meets the system requirements.
If it does, you can get the free upgrade when it's available.

Backup your apps, settings, and files so they're available across devices. Open Windows Backup

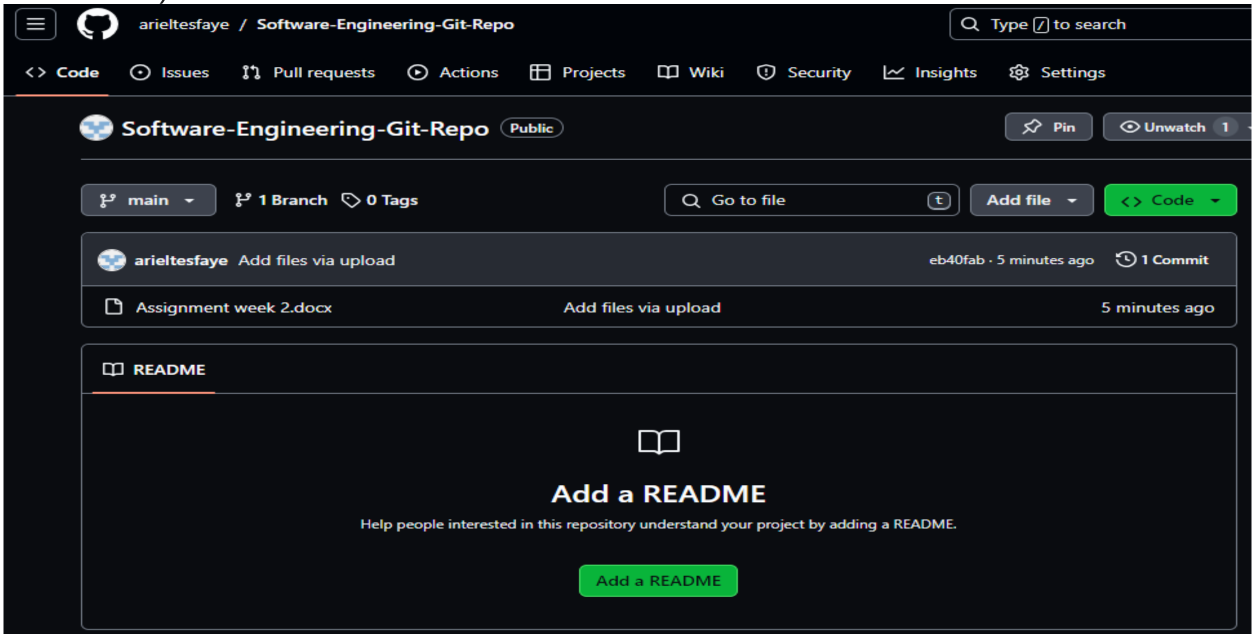
2. Installing Text Editor/IDE:

Visual Studio Code (VS Code) is a popular and versatile IDE. We'll proceed with installing VS Code).


File Edit Selection View ···
Search
Welcome
Visual Studio Code
Editing evolved
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Recent
Week 6 C:\Users\hp240\Desktop\PLP Lessons\DART
Week 4 & 5 Object Oriented Programming C:\Users\hp240\...
PersonalPortfolioWebsite C:\Coding\Web Development VS c...
Class C:\Coding\Python
30 day challenge C:\Coding\Python
More...
Videos
Watch Getting Started Tutorials
Learn VS Code's must-have features in short and practical videos
Walkthroughs
Get Started with VS Code
Learn the Fundamentals
Get Started with Python Developm... Updated
Get Started with C++ Development Updated
Browse & Edit Remote Repositories... Updated
More...
⊗ 0 ⚠ 0 0

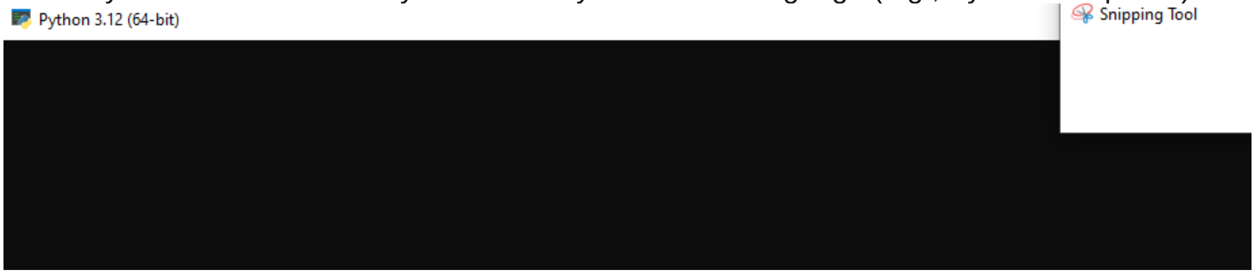
3. Setting Up Version Control System:

- Create a **GitHub account** (online platform for hosting code repositories)
- Initialize a Git repository for your project and make your first commit (saving a snapshot of your code's state).



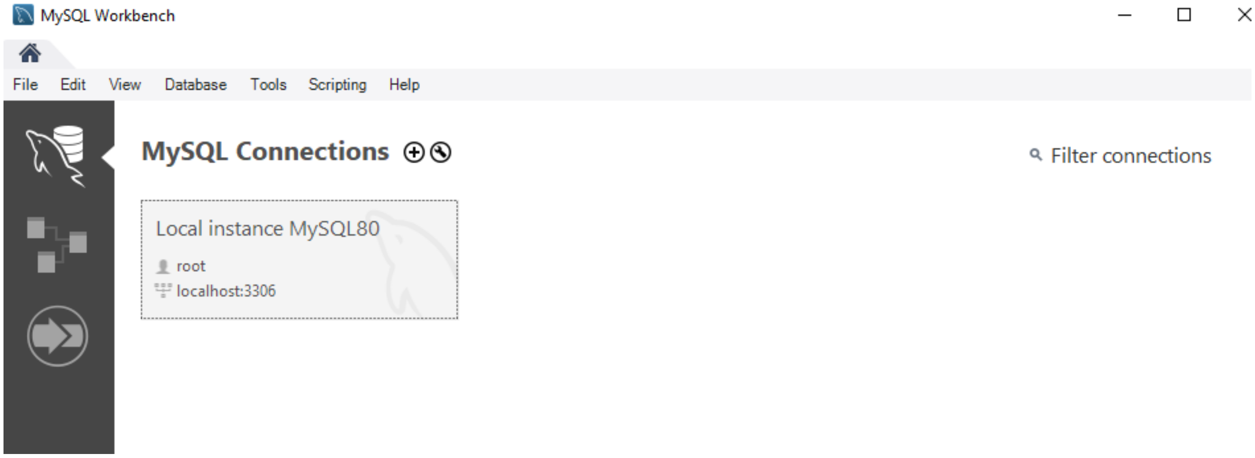
4. Installing Necessary Programming Languages and Runtimes:

- Download and install the programming language your project requires (e.g., Python from http://www.python.org).
- Ensure you have the necessary tools to run your chosen language (e.g., Python interpreter).



6. Configuring a Database (Optional):

- Download and install **MySQL**



8. Explore Extensions and Plugins:

- Explore available extensions, plugins, and add-ons for VS Code (or your chosen editor/IDE).
- Examples include syntax highlighting (color coding for different code elements), code formatting (automatic formatting for readability), and version control integration (easier interaction with Git).

Assignment 2

1. **Describe the steps to download and install Visual Studio Code on Windows 11 operating system. Include any prerequisites that might be needed.**

   i. **Prerequisites**: Ensure that your system meets the minimum requirements for running Visual Studio Code. This typically includes having Windows 11 installed with up-to-date system updates.

   ii. **Download Visual Studio Code:** Open your web browser and navigate to the official Visual Studio Code website at https://code.visualstudio.com/. Click on the "Download for Windows" button to download the installer.

   iii. **Run the Installer**: Once the installer has finished downloading, locate the downloaded file (usually in your Downloads folder) and double-click on it to run the installer.

   iv. **Installation Process**: Follow the on-screen instructions provided by the installer. You may be prompted to choose installation options such as the installation location and whether to add shortcuts to the Start Menu or desktop.

   v. **Completing the Installation**: After the installation process completes, you can launch Visual Studio Code by double-clicking its icon on the desktop or finding it in the Start Menu.

   vi. **Configure Visual Studio Code**: Upon launching Visual Studio Code for the first time, you may customize its settings and install extensions according to your preferences and development needs.

2. **After installing VS Code, what initial configurations and settings should be adjusted for an optimal coding environment? Mention any important settings or extensions.**

   a. Theme: Choose a theme that's easy on the eyes for long coding sessions.

   b. Font: Select a comfortable font.

   c. Editor Settings: These settings are usually found in the editor settings (`settings.json`).

   d. Extensions:
      i. Language Support: Install extensions for the programming languages you'll be using.

ii. Version Control
iii. Debugging: Debugger for Chrome for debugging JavaScript in Chrome.
iv. Code Formatting: Prettier for automatic code formatting.
v. Code Snippets: Install snippets for your preferred languages for faster coding.
vi. File Icons: Icons for different file types for better visual organization.

e. Integrated Terminal: Set up the integrated terminal with your preferred shell (e.g., PowerShell, Bash).

f. Linting and Error Checking: Configure ESLint, TSLint, or any other linter for your project's language.

g. Workspace Settings: Adjust settings specific to your workspace, such as file associations and project-specific configurations.

3. **Explain the main components of the VS Code user interface. Identify and describe the purpose of the Activity Bar, Side Bar, Editor Group, and Status Bar.**

i. Activity Bar: Located on the side, it provides quick access to different views like Explorer, Source Control, and Extensions.

ii. Side Bar: Contains different panels like Explorer for file navigation, Search for searching within files, and Extensions for managing installed extensions.

iii. Editor Group: Holds one or more editors where you can view and edit files. Each group can display different files or split views.

iv. Status Bar: Located at the bottom, it displays information like line and column number, file encoding, and language mode. It also includes features like language mode selection and indentation settings.

4. **What is the Command Palette in VS Code, and how can it be accessed? Provide examples of common tasks that can be performed using the Command Palette.**

The Command Palette in VS Code is a tool accessed via `Ctrl+Shift+P`. It allows users to execute various commands, such as opening files, searching for symbols, running tasks, and installing extensions, by typing commands or selecting from a list. Examples of tasks include changing the color theme, running Git commands, and managing extensions.

5. **Discuss the role of extensions in VS Code. How can users find, install, and manage extensions? Provide examples of essential extensions for web development.**

   Extensions in VS Code enhance its functionality by adding features like language support, debugging tools, and themes. Users can find, install, and manage extensions through the Extensions view (`Ctrl+Shift+X`). Essential extensions for web development include:
   Live Server: For live server functionality.
   ESLint: For JavaScript linting.
   Prettier: For code formatting.
   Debugger for Chrome: For debugging JavaScript in Chrome.
   HTML CSS Support: For improved HTML and CSS support.

6. **Describe how to open and use the integrated terminal in VS Code. What are the advantages of using the integrated terminal compared to an external terminal?**

   To open the integrated terminal in VS Code, use `Ctrl+`` (Backtick) or navigate to `View` > `Terminal`. Once opened, you can use it like any other terminal to run commands, execute scripts, and manage your project.

   Advantages of the integrated terminal:
   - Seamless Integration: No need to switch between different applications.
   - Contextual Awareness: Automatically opens to the project directory, providing easy access to project-specific commands.
   - Customization: Can be configured with preferred shell, fonts, and colors.
   - Productivity: Saves time and streamlines workflow by keeping everything within the editor.

7. **Explain how to create, open, and manage files and folders in VS Code. How can users navigate between different files and directories efficiently?**

   To create a new file or folder in VS Code, use the Explorer view (`Ctrl+Shift+E`) and right-click in the file tree area. To open files, simply double-click them in the Explorer or use `Ctrl+P` to search for and open specific files by name.

   To manage files and folders:
   - Right-click to access options like renaming, deleting, or copying.
   - Drag and drop files to rearrange or move them.

   To navigate efficiently between files and directories:
   - Use breadcrumbs at the top of the editor to quickly jump between directories.

- Utilize the Go to File (`Ctrl+P`) or Go to Symbol (`Ctrl+Shift+O`) commands to search for files or symbols within files.

8. **Where can users find and customize settings in VS Code? Provide examples of how to change the theme, font size, and key bindings.**

   Users can find and customize settings in VS Code by accessing the Settings view (`Ctrl+,`).

   Examples of customizations:
   - To change the theme, search for "Color Theme" in settings and select the desired theme.
   - To adjust the font size, search for "Font Size" in settings and modify the value accordingly.
   - To customize keybindings, search for "Keybindings" in settings and click on "Edit keybindings.json" to modify keybindings manually or use the UI to reassign keybindings.

9. **Outline the steps to set up and start debugging a simple program in VS Code. What are some key debugging features available in VS Code?**
   Setting Up:

   i. **Create/Open Program:** Write your code in the appropriate file extension (e.g., .py for Python, .js for JavaScript) or open an existing program.
   ii. **Enable Debugging:** Go to the "Run and Debug" view (Ctrl+Shift+D or click the debug icon) and select "Run and Debug" (or the specific configuration for your language).

   Debugging Process:

   i. **Set Breakpoints:** Click next to the line of code where you want to pause execution. A red dot signifies the breakpoint.
   ii. **Start Debugging:** Click the "Run" button (F5) to start debugging. Execution halts at the first breakpoint.

   Key Features:

   - **Breakpoints:** Pause execution for inspection.
   - **Call Stack:** View the sequence of function calls for understanding execution flow.
   - **Variables:** Inspect values of variables at breakpoints to verify data correctness.
   - **Step Commands:**
     - **Step Over:** Executes the current line without entering functions.
     - **Step Into:** Enters a function call, stepping through its code.
     - **Step Out:** Exits the current function, returning to the caller.
   - **Console:** View program output or interact with the program using the built-in console.

10. **How can users integrate Git with VS Code for version control? Describe the process of initializing a repository, making commits, and pushing changes to GitHub.**
    Prerequisites:

- Git installed: Download from https://www.git-scm.com/downloads

  Steps:

  i. **Initialize Repository:** Open a terminal in VS Code (Terminal > New Terminal). Navigate to your project directory (e.g., cd Documents/MyProject). Run git init to create a new Git repository.

  ii. **Create GitHub Account:** If you don't have one, sign up for a free account at https://github.com/index.

  iii. **Connect VS Code (Optional):** Search "Git: Authentication" in VS Code settings (Ctrl+,) and configure Git credentials for easier pushing.

  iv. **Stage Changes:** Use the Source Control view to view changed files. Select files for the next commit and click "Stage Changes" (or right-click menu).

  v. **Make a Commit:** In the Source Control view, click "+" or type a commit message, then press Ctrl+Enter. This creates a snapshot of your project's state.

  vi. **Push to GitHub:**

     a. In the terminal, navigate to your project directory.

     b. Run git remote add origin https://github.com/[Your Username]/[Your Repository Name].git to link your local repository to your remote repository on GitHub (replace placeholders).

     c. Run git push -u origin master to push your local commits to the master branch on your GitHub repository. -u sets the upstream branch for simpler future pushes.