[![Review Assignment Due Date](https://classroom.github.com/assets/deadline-readme-button-22041afd0340ce965d47ae6ef1cefeee28c7c493a6346c4f15d667ab976d596c.svg)](https://classroom.github.com/a/XoLGRbHq)

[![Open in Visual Studio Code](https://classroom.github.com/assets/open-in-vscode-2e0aaae1b6195c2367325f4f02e2d04e9abb55f0b24a779b69b11b9e10269abc.svg)](https://classroom.github.com/online\_ide?assignment\_repo\_id=15278794&assignment\_repo\_type=AssignmentRepo)

**# SE-Assignment-5**

Installation and Navigation of Visual Studio Code (VS Code)

 Instructions:

Answer the following questions based on your understanding of the installation and navigation of Visual Studio Code (VS Code). Provide detailed explanations and examples where appropriate.

 Questions:

1. Installation of VS Code:

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

   Installation of Visual Studio Code on Windows 11

1. Prerequisites:

1. Administrative Rights: Ensure you have administrative rights to install software on your computer.

2. Internet Connection: An active internet connection to download the installer.

1) Steps to Download and Install Visual Studio Code:

2) Download VS Code Installer:

a) Open your web browser and navigate to the official Visual Studio Code website.

b) On the homepage, you will see a download button labeled "Download for Windows". Click this button to download the VS Code installer.

3) Run the Installer:

a) Once the download is complete, locate the installer file (e.g., VSCodeSetup-x64-1.x.x.exe) in your Downloads folder.

b) Double-click the installer file to start the installation process.

4) Setup Wizard:

a) The Visual Studio Code Setup Wizard will open. Click "Next" to proceed through the steps.

b) License Agreement: Read the license agreement. If you agree to the terms, select the checkbox "I accept the agreement" and click "Next".

5) Select Destination Location:

a) Choose the folder where you want to install Visual Studio Code. The default location is usually fine. Click "Next" to continue.

6) Select Additional Tasks:

a) Choose any additional tasks you want to perform during the installation. Common options include:

i) Creating a desktop icon.

ii)   Adding "Open with Code" action to Windows Explorer file context menu.

iii)  Adding "Open with Code" action to Windows Explorer directory context menu.

iv)   Registering VS Code as an editor for supported file types.

v) Adding to the PATH (important for using VS Code from the command line).

7) Ready to Install:

a) After selecting your preferences, click "Next" and then "Install" to begin the installation.

8) Complete Installation:

a) Once the installation is complete, you can choose to launch Visual Studio Code immediately by selecting the checkbox "Launch Visual Studio Code" and then clicking "Finish".

1) Post-Installation:

First-Time Setup:

a) When you first open VS Code, you may be prompted to customize your setup. You can select a theme, install recommended extensions, and configure settings as needed.

b) You may also be prompted to sign in with a GitHub or Microsoft account to sync your settings and extensions across devices.

2. First-time Setup:

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

   Initial Configurations and Settings

1. Theme and Font Settings:

I. Theme: Choose a theme that suits your preference for better readability and comfort.

♣  Go to File > Preferences > Color Theme and select from the available themes.

II.   Font: Adjust the font family and size for better clarity.

♣  Go to File > Preferences > Settings and search for editor.fontFamily and editor.fontSize.

2. Tab and Indentation Settings:

I. Indentation: Configure the tab size and ensure consistent use of spaces or tabs.

♣  Go to File > Preferences > Settings and search for editor.tabSize and editor.insertSpaces.

II.   Auto Format: Enable auto-format on save for cleaner code.

♣  Search for editor.formatOnSave and enable it.

3. File Encoding:

I. Set the default file encoding to UTF-8.

♣  Go to File > Preferences > Settings and search for files.encoding and set it to utf8.

4. Word Wrap:

I. Enable word wrap for better readability of long lines of code.

♣  Go to File > Preferences > Settings and search for editor.wordWrap and set it to on.

Recommended Extensions

1. Code Formatting and Linting:

I. Prettier: An opinionated code formatter that supports many languages.

♣  Install the Prettier extension from the Extensions view (Ctrl+Shift+X).

♣  Configure it by adding "editor.defaultFormatter": "esbenp.prettier-vscode" in your settings.

II.   ESLint: A linter for identifying and reporting on patterns found in ECMAScript/JavaScript code.

♣  Install the ESLint extension and configure it with a .eslintrc file in your project.

2. Version Control:

I. GitLens: Enhances the built-in Git capabilities, providing insights into code authorship and history.

♣  Install GitLens from the Extensions view.

3. Language Support:

I. Python: Provides rich support for the Python language, including features like IntelliSense, linting, and debugging.

♣  Install the Python extension.

II.   JavaScript/TypeScript: For better JavaScript and TypeScript support, ensure you have the official extensions installed and configured.

4. Snippets:

I. Code Snippets: Various extensions provide code snippets for different languages.

♣  Install the relevant snippets extensions for your preferred languages, such as JavaScript (ES6) code snippets.

5. Live Server:

I. Live Server: Launch a local development server with live reload feature for static and dynamic pages.

♣  Install the Live Server extension and use it to preview your HTML, CSS, and JavaScript changes in real-time.

3. User Interface Overview:

   - 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.

   1. Activity Bar

•  Located on the left-hand side of the window.

•  Provides quick access to frequently used features such as:

1. File Explorer

2. Debugger

3. Task Manager

4. Terminal

•  Allows users to switch between different views and tools easily.

2. Side Bar

•  Located within the Activity Bar.

•  Contains the File Explorer and other context-sensitive views.

•  Provides a hierarchical view of the project's files and folders.

•  Allows users to navigate, create, and modify files.

3. Editor Group

•  Occupies the central area of the window.

•  Contains one or more editor tabs.

•  Each tab represents an open file.

•  Provides a space for editing, viewing, and debugging code.

4. Status Bar

•  Located at the bottom of the window.

•  Displays important information such as:

ϖ  Current line and column number

ϖ  Language mode

ϖ  Encoding type

ϖ  Git status

ϖ  Error and warning messages

•  Provides quick feedback and helps users monitor the state of their project.

4. Command Palette:

   - 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.

How to Access the Command Palette

The Command Palette can be accessed in two main ways:

Using Keyboard Shortcuts:

Press Ctrl+Shift+P to open the Command Palette.

Using the Menu:

Go to the View menu and select Command Palette.

Examples of Common Tasks Performed Using the Command Palette

Changing the Color Theme:

Open the Command Palette with Ctrl+Shift+P.

Type theme and select Preferences: Color Theme.

Choose from the list of available themes to change the color scheme of the editor.

Running a Build Task:

Open the Command Palette with Ctrl+Shift+P.

Type task and select Tasks: Run Build Task.

This runs the configured build task for the current project.

Installing Extensions:

Open the Command Palette with Ctrl+Shift+P.

Type install and select Extensions: Install Extensions.

Search for the desired extension and install it directly from the Command Palette.

Opening Settings:

Open the Command Palette with Ctrl+Shift+P.

Type settings and select Preferences: Open Settings (UI) to open the settings interface.

Navigating to a File:

Open the Command Palette with Ctrl+Shift+P.

Type open file and select File: Open File....

Browse and open the desired file within the workspace.

Git Commands:

Open the Command Palette with Ctrl+Shift+P.

Type git to see various Git commands like Git: Commit, Git: Push, Git: Pull, etc.

Format Document:

Open the Command Palette with Ctrl+Shift+P.

Type format and select Format Document.

This formats the current document using the configured formatter.

Toggle Terminal:

Open the Command Palette with Ctrl+Shift+P.

Type terminal and select Terminal: Toggle Integrated Terminal.

This opens or closes the integrated terminal within VS Code.

View and Edit Keyboard Shortcuts:

Open the Command Palette with Ctrl+Shift+P.

Type shortcuts and select Preferences: Open Keyboard Shortcuts.

View and customize the keyboard shortcuts for various commands.

Create a New File:

Open the Command Palette with Ctrl+Shift+P.

Type new file and select File: New File.

This creates a new file in the current workspace.

5. Extensions in VS Code:

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

   1. Role of Extensions in VS Code

1. Language Support: Syntax highlighting, code completion, linting, and debugging for various programming languages.

2. Theming: Custom color themes and icons to personalize the look and feel of the editor.

3. Development Tools: Integration with build tools, version control systems, and task runners.

4. Utilities: Additional functionalities like snippets, live server, and project management tools.

2. Finding, Installing, and Managing Extensions

1. Finding Extensions:

o  VS Code Marketplace: The primary source for discovering and installing extensions. It can be accessed online at Visual Studio Code Marketplace or within VS Code itself.

o  Within VS Code:

♣  Open the Extensions view by clicking on the Extensions icon in the Activity Bar on the side of the window or by pressing Ctrl+Shift+X.

2. Installing Extensions:

o  Using the Extensions View:

♣  Search for the desired extension using the search bar.

♣  Click the Install button next to the extension name.

o  Using the Command Palette:

♣  Press Ctrl+Shift+P to open the Command Palette.

♣  Type ext install followed by the name of the extension and select it from the list to install.

3. Managing Extensions:

o  Disabling/Enabling Extensions:

♣  Right-click on an installed extension in the Extensions view and select Disable or Enable.

o  Uninstalling Extensions:

♣  Click on the Uninstall button next to the extension name in the Extensions view.

o  Updating Extensions:

♣  Extensions can be updated automatically, or you can manually check for updates in the Extensions view by clicking on the Update button if available.

3. Extensions for Web Development

1. ESLint:

o  Provides JavaScript and TypeScript linting.

2. Prettier - Code Formatter:

o  Automatically formats your code to adhere to specified style guidelines.

3. Live Server:

o  Launches a local development server with live reload feature for static and dynamic pages.

4. Debugger for Chrome:

o  Allows debugging JavaScript code in the Google Chrome browser.

5. HTML Snippets:

o  Provides a collection of useful HTML code snippets.

6. CSS Peek:

o  Allows you to easily find CSS definitions in your project.

o  Useful for navigating between HTML and CSS files.

7. Path Intellisense:

o  Helps to avoid mistakes when typing paths to files.

8. GitLens:

o  Enhances the built-in Git capabilities in VS Code.

6. Integrated Terminal:

   - 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?

   1. Opening the Integrated Terminal

1. Opening the Integrated Terminal:

•  Press Ctrl+` (backtick) to open the integrated terminal. Alternatively, you can go to View > Terminal from the menu, or use the command palette (Ctrl+Shift+P) and type "Terminal: Toggle Terminal".

2. Switching Terminal Shells:

•  By default, VS Code opens the integrated terminal with the default shell configured on your system (e.g., Bash on Linux/macOS, Command Prompt on Windows). You can change the default shell by clicking on the dropdown arrow in the terminal panel and selecting a different shell if needed.

3. Using the Terminal:

•  Once the terminal is open, you can type commands directly into it as you would in an external terminal. Use standard shell commands to navigate directories, run scripts, install dependencies (npm install, pip install, etc.), and execute build processes.

4. Terminal Navigation Shortcuts:

•  Use standard keyboard shortcuts within the integrated terminal:

•  Ctrl+C: Interrupt (SIGINT) the currently running process.

•  Ctrl+D (Linux/macOS) or Ctrl+Z (Windows): Exit the current shell session.

•  Up Arrow and Down Arrow: Navigate through command history.

•  Tab: Auto-completion of file names and commands.

5. Managing Multiple Terminals:

•  You can open multiple terminal instances within VS Code by clicking on the + button in the terminal panel or by using the command palette to create new terminals (Terminal: Create New Integrated Terminal).

   2. Using the Integrated Terminal

The integrated terminal provides a command-line interface within your code editor. You can use it to:

1. Run commands and scripts

2. Interact with the terminal emulator (e.g., bash, PowerShell)

3. Monitor build and debug output

4. Access file system and environment variables

5. Perform version control operations (e.g., git)

   3. Advantages of the Integrated Terminal

Compared to an external terminal, the integrated terminal in VS Code offers several advantages:

1. Convenience: Easily accessible from within the editor, eliminating the need to switch between windows.

2. Integration: Runs within the VS Code environment, allowing for seamless interaction with code, plugins, and extensions.

3. Customization: Supports customization with themes, fonts, and other settings to enhance user experience.

4. Autocompletion: Provides autocompletion for commands and parameters, improving efficiency.

5. Multilingual: Supports multiple terminal emulators (e.g., bash, PowerShell, Zsh) to cater to different preferences.

6. Task Automation: Integrates with tasks (triggered by keybindings or code changes) for automated execution of commands within the terminal.

7. Debug Support: Facilitates debugging of terminal commands and scripts within the VS Code debugger.

7. File and Folder Management:

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

   1. Creating Files and Folders

1. Creating a New File:

o  To create a new file, you can use either the file explorer sidebar or the command palette (Ctrl+Shift+P).

♣  Sidebar: Right-click on the folder where you want to create the file, then select New File.

♣  Command Palette: Open the command palette (Ctrl+Shift+P), type "New File", and select File: New File. Enter the file name when prompted.

  2.  Creating a New Folder:

o  Similarly, use the file explorer sidebar or command palette to create a new folder.

♣  Sidebar: Right-click on the parent folder, then select New Folder.

♣  Command Palette: Open the command palette, type "New Folder", and select File: New Folder. Enter the folder name when prompted.

2. Opening Files and Folders

1. Opening Files:

o  To open an existing file, navigate to the file explorer sidebar.

♣  Sidebar: Click on the file name to open it in the editor area.

♣  Command Palette: Open the command palette, type "Open File", and select File: Open File. Enter the file path or select from the recent files list.

2. Opening Folders:

o  Open entire folders in VS Code for comprehensive project management.

♣  Sidebar: Click on File > Open Folder... and select the folder you want to open.

♣  Command Palette: Type "Open Folder" in the command palette and select File: Open Folder. Choose the folder from the file explorer.

   3. Managing Files and Folders

1. Renaming Files and Folders:

o  Sidebar: Right-click on the file or folder, select Rename, and enter the new name.

o  Command Palette: Type "Rename" in the command palette and select File: Rename File. Enter the new name.

2. Deleting Files and Folders:

o  Sidebar: Right-click on the file or folder, select Delete, and confirm the action.

o  Command Palette: Type "Delete" in the command palette and select File: Delete File or File: Delete Folder.

3. Moving/Copying Files and Folders:

o  Use the file explorer sidebar to drag and drop files/folders to move or copy them within or across directories.

   4. Navigating Between Files and Directories Efficiently

1. Switching Between Open Files:

o  Use Ctrl+Tab to cycle through recently opened files.

o  Use Ctrl+P to open the Quick Open dialog and type the file name to navigate directly.

2. Navigating in the File Explorer:

o  Use the file explorer sidebar to browse and navigate between different folders and files.

o  Collapse or expand folders by clicking on the arrow next to the folder name.

3. Using the Go to Symbol Feature:

o  Press Ctrl+Shift+O to open the Go to Symbol dialog. Type to filter and navigate directly to functions, classes, or symbols within the current file.

4. Navigating by File Paths:

o  Use Ctrl+Shift+E to focus on the file explorer sidebar and navigate by typing file paths.

o  Use Ctrl+Shift+F to search across all files in the workspace by content or filename.

   5. Tips for Efficiency

1. Use Shortcuts: Memorize and use keyboard shortcuts for common actions like creating new files (Ctrl+N), opening files (Ctrl+P), and saving changes (Ctrl+S).

2. Customize Sidebar: Customize the file explorer sidebar to show only relevant folders or hide unnecessary ones.

3. Split Editor: Use Ctrl+\ to split the editor into multiple panes, allowing simultaneous editing and reference.

4. Extensions: Install extensions like Project Manager or File Utils to enhance file and folder management capabilities.

8. Settings and Preferences:

   - Where can users find and customize settings in VS Code? Provide examples of how to change the theme, font size, and keybindings.

   1. Settings Menu:

•  Click on the File menu and select Preferences (MacOS) or Settings (Windows).

2. Settings Editor:

•  A new tab will open displaying the Settings Editor.

•  Search for specific settings using the search bar.

•  Alternatively, browse through the categories on the left sidebar.

3. Changing the Theme:

•  Search for "theme" in the Settings Editor.

•  Click on the Color Theme dropdown and select a desired theme.

•  Example: {"workbench.colorTheme": "Dracula"}

4. Changing the Font Size:

•  Search for "editor.fontSize" in the Settings Editor.

•  Enter the desired font size in pixels.

•  Example: {"editor.fontSize": 14}

5. Changing Keybindings:

•  Search for "keybindings" in the Settings Editor.

•  Click on the Open Keyboard Shortcuts (JSON) button.

•  Locate the keybinding you want to change.

•  Edit the key value to a new key combination.

•  Example: {"key": "ctrl+k c"} for copying text.

9. Debugging in VS Code:

   - 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?

Steps to Set Up and Start Debugging in VS Code:

1. Install the Debugger Extension: Go to the VS Code Marketplace and install the "Debugger for Chrome" or "Debugger for Node.js" extension, depending on the language you're using.

2. Open the Debug View: Click on the "Debug" icon in the sidebar or press "Ctrl + Shift + D" to open the Debug view.

3. Create a Debug Configuration File: In the "Run and Debug" section, click on the "Create a launch.json file" link. This file will store the configuration settings for your debugging session.

4. Define a Configuration: In the launch.json file, specify the following settings:

type: The type of debugging session (e.g., "node", "chrome")

request: The type of debugging request (e.g., "launch", "attach")

name: A name for the configuration

program: The path to the program you want to debug

5. Start Debugging:

Launch: Click on the green "play" button next to the configuration name in the Debug view to launch the program in debug mode.

Attach: Click on the "attach" button to attach to an already running program.

10. Using Source Control:

    - 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.

1.   Initialize a Repository:

•  Click Initialize Repository.

2.  Stage Changes:

•  Click + next to modified files to stage them.

3. Commit Changes:

•  Enter a commit message and click the checkmark.

4.  Add Remote:

•  In the terminal: git remote add origin https://github.com/username/repository.git.

5. Push to GitHub:

•  In the Source Control view, click Push.

 Submission Guidelines:

- Your answers should be well-structured, concise, and to the point.

- Provide screenshots or step-by-step instructions where applicable.

- Cite any references or sources you use in your answers.

- Submit your completed assignment by 1st July