Installation and Navigation of Visual Studio Code (VS Code)

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.

Setting up Visual Studio involves a few steps. Here's a concise step-by-step guide:

i) Download Visual Studio:

Visit the Visual Studio website.
Click the "Download for Windows" button. This will download the installer file.
#

ii) Run the installer:

Once the download is complete, locate the downloaded file (usually in your Downloads folder).
Double-click the installer to begin the installation process.
Choose the “Visual Studio” workload during installation, which includes the necessary components for general development.

iii) Select Workloads and Components:

The installer will launch the setup wizard. You'll be presented with the license agreement. Proceed by clicking “Accept” to continue.
In the Visual Studio Installer, select the workloads and components you need based on your development requirements. Common workloads include ".NET Desktop Development" or "Web Development."

iv) Complete the installation:

Once you've reviewed the options, click "Next" to proceed with the installation.
Click the “Install” button to start the installation process.

The installer will copy the necessary files and configure VS Code on your system. This might take a minute or two.

v) Launch VS Code:

Once the installation is complete, launch Visual Studio.
Sign in with your Microsoft account or create one if prompted.

2. First-time Setup

After installing VS Code the initial configuration and settings that should be adjusted depend on the programming languages you use.

General settings

Theme: VS Code offers a variety of light and dark themes. Choose one that suits your preference. You can find them under **File > Preferences > Settings**. Search for "Theme".

Font-size and Style: Adjust the font size and style for better readability. Navigate to **File > Preferences > Settings** and search for "Font Size" and "Font Family."

Keyboard Shortcuts: VS Code offers a range of built-in keyboard shortcuts. Search for "Keyboard Shortcuts" in the settings to explore and modify them.

Extensions

Language-specific extensions: These extensions provide syntax highlighting, code completion, debugging, and other functionalities for specific languages like Python (Pylance), C++ (C/C++), or Java (Java Extension Pack).

Git integration: GitHub Pull Requests enhance your Git workflow within VS Code, allowing you to visualize code history and manage pull requests directly.

Productivity extensions: Extensions like Bracket Pair Colorizer or Code Runner improve code readability and allow you to execute code snippets within VS Code.

Configurations

Configure the integrated terminal to use a terminal of your choice as default like PowerShell, Git Bash or Command Prompt.

3. User Interface Overview

Editor: This is the central area where you write and edit your code. You can open multiple files simultaneously and arrange them in tabs or split views for efficient code comparison and editing.

Activity Bar: This vertical bar provides quick access to different VS Code functionalities. It typically includes icons for:

- **Explorer:** Manage your project files and folders.
- **Search:** Search for text across your codebase.
- **Source Control (Git):** View and manage your Git repositories.
- **Run and Debug:** Configure debugging options and launch your application.
- **Extensions:** Manage installed extensions and discover new ones.

Side Bar: This area displays different views depending on the context you're working in. Some common views include:

- **Explorer:** Provides a file tree view of your project.
- **Search results:** Lists search matches within your codebase.
- **Source Control view:** Shows Git status of your files and lets you commit changes.
- **Output:** Displays messages from the terminal, debugger, or other tools.

Status Bar: This bar displays information about the current file, like its language, line and column position, indentation mode, and encoding. It can also show Git status and provide quick actions depending on the context.

4. Command Palette:

The Command Palette in Visual Studio Code is a powerful search bar that allows you to quickly access all functionalities within the editor, keyboard shortcuts included. It's a central hub for finding and executing various commands without navigating through menus. The most common way of access it is by pressing `Ctrl+Shift+P`.

Examples of Common Tasks with the Command Palette:

- Open Files
- Search for Symbols
- Navigate the Interface

- Install Extensions

5. Extensions in VS Code

Extensions are essentially add-ons that contribute new features and functionalities to VS Code, allowing you to tailor the editor to your specific programming needs and preferences.

Finding, Installing, and Managing Extensions

 # Access the Extensions View: Open the **Extensions** view by clicking on the Extensions icon (puzzle piece icon) in the Activity Bar on the left, or navigate to **View > Extensions**.

Explore the Marketplace: The Extensions view displays a search bar and a curated list of popular extensions. Search for extensions by keyword or browse by category.
Install and Manage: Click on the desired extension to view its details, ratings, and functionalities. Click the "Install" button to add the extension to your VS Code setup. Once installed, you can manage extensions from this view as well, enabling or disabling them as needed.

Essential Extensions for Web Development

Language-specific extensions:

- **HTML, CSS, JavaScript (IntelliSense):** Provides intelligent code completion, syntax highlighting, and debugging support for these core web development languages.

- **React (JavaScript extension with React support):** Offers features specific to React development, like component snippets, prop suggestions, and debugging tools.

- **Angular (Angular Language Service):** Enhances your Angular development experience with code completion, navigation, and linting specific to the Angular framework.

Linters and Formatters:

- **ESLint:** Identifies potential errors and stylistic issues in your JavaScript code, enforcing consistent formatting and best practices.

- **Prettier:** Automatically formats your code according to a defined style guide, ensuring consistent formatting across your project.

Productivity Boosters:

- **Live Server:** Launches a local development server with live reload functionality, automatically refreshing the browser whenever you save your code.

- **GitLens:** Supercharges your Git workflow within VS Code, providing visual Git history, blame annotations, and easier repository navigation.

- **Bracket Pair Colorizer:** Improves code readability by color-coding matching brackets and parentheses, making it easier to identify code blocks.

Debuggers:

- **Debugger for Chrome/Firefox:** Enables debugging your web applications directly within VS Code, allowing you to set breakpoints, inspect variables, and step through code execution.

6. Integrated Terminal

Opening the Terminal:

There are two ways to open the integrated terminal:

- **Keyboard Shortcut:** The most common way is by pressing `Ctrl+`` (backtick) on Windows/Linux or `Cmd+`` (backtick) on Mac.

- **Menu:** Navigate to the **View > Terminal** menu option.

Advantages of Using the Integrated Terminal:

There are several advantages to using the integrated terminal within VS Code compared to opening a separate terminal window:

- **Convenience:** Switching between code editing and the terminal is much faster and more seamless as you don't need to leave the VS Code window.

- **Working Directory:** The terminal automatically opens with the working directory set to your current project folder, saving you the need to navigate there manually.

- **Split View:** You can split the editor window to have your code alongside the terminal, allowing you to refer to both simultaneously.

- **Command Palette Integration:** You can search for and run terminal commands directly from the Command Palette (Ctrl+Shift+P) within VS Code.

- **Customization:** You can customize the look and feel of the integrated terminal, including the font size, colors, and theme, to match your preferences.

7. File and Folder Management

Creating Files and Folders:

- **Right-Click Menu:** Navigate to the desired location in the Explorer view, right-click on an empty space, and select "New File" or "New Folder" from the context menu.

- **Keyboard Shortcuts:** Use `Ctrl+N` (Windows/Linux) or `Cmd+N` (Mac) to open a "New File" dialog, or `Ctrl+Shift+N` (Windows/Linux) or `Cmd+Shift+N` (Mac) for a "New Folder" dialog.

Opening files:

- **Double-Click:** Double-clicking on a file name in the Explorer view opens it in the editor window.

- **Search Bar:** The Explorer view has a search bar at the top. Type the name of the file you want to open, and VS Code will locate and display it.

Managing files and folders:

- **Renaming:** Click once on a file or folder name to highlight it, then click again to enter renaming mode. Press Enter to save the new name.

- **Deleting:** Right-click on a file or folder and select "Delete" from the context menu.

- **Moving and Copying:** Use drag-and-drop functionality within the Explorer view to move or copy files and folders between locations within your project. You can also use the right-click menu options like "Cut" and "Paste" for moving, or "Copy" and "Paste" for copying.

Navigating Efficiently:

- **File Tabs:** As you open multiple files, VS Code displays them as tabs along the top of the editor area. Clicking on a tab switches to the corresponding file.

- **Go to File:** Use `Ctrl+T` (Windows/Linux) or `Cmd+T` (Mac) to open the "Go to File" dialog. Start typing the name of the file you want to switch to, and VS Code will suggest matching files for quick selection.

- **Open Recent:** Navigate to the **File > Open Recent** menu to see a list of recently opened files for easy access.

- **Split View:** Drag a file from the Explorer view and drop it next to another open file to create a split view, allowing you to see both files side-by-side.

8. Settings and Preferences

Accessing Settings:

- **Menu:** Navigate to **File > Preferences > Settings** (or **Code > Preferences > Settings** on macOS).

- **Keyboard Shortcut:** Press `Ctrl+,` (comma) on Windows/Linux or `Cmd+,` (comma) on Mac.

Changing Theme:

- In the search bar, type "Theme". This will display all theme-related settings.
- Under the "Color Theme" setting, you'll see a dropdown menu with available themes. Select the theme you prefer (e.g., Dark+, Light Theme).

Adjusting Font Size:

- Search for "Font Size" in the settings bar.
- You'll see an option labeled "Editor Font Size". Enter a specific pixel value to adjust the font size to your liking.

Customizing Keybindings:

- Search for "Keyboard Shortcuts" in the settings bar.
- VS Code displays a default keybinding settings file. You can modify these bindings directly or choose to open the settings.json file for more advanced customization.
- The settings.json file allows you to define custom keyboard shortcuts for specific actions or functionalities within VS Code.

9. Debugging in VS Code:

Open the Program in VS Code:

- navigate to it within VS Code's Explorer view and open it in the editor.

Set Breakpoints:

- Click on the line number of your code where you want execution to pause during debugging. A red dot will appear indicating the breakpoint.

Start Debugging:

- Click the "Run and Debug" button (play icon) in the debug control bar (or press F5).

Key Debugging Features in VS Code:

- **Breakpoints:** Set pause points in your code for inspection.
- **Step Execution:** Control program execution line by line.
- **Variable Inspection:** View and modify variable values during debugging.
- **Call Stack Exploration:** See the call history and function nesting.
- **Conditional Breakpoints:** Set breakpoints that only trigger under specific conditions.
- **Integrated Console:** View logs, errors, and interact with the program during debugging.

10. Using Source Control:

Initialize a Git Repository:

- Open your project folder in VS Code.

- Go to the Source Control view (usually on the left sidebar) or click the Git icon in the status bar (bottom right corner).

- Click the “+” button and select “Initialize Repository”. This creates a new Git repository within your project folder.

Stage Changes for Commits:

- Make changes to your code files

- In the Source Control view, you'll see modified files with a colored icon (usually green). These represent unstaged changes.

- Click the "+" icon next to the file name to stage the changes for your next commit. Stages changes are included in the next commit.

Commit Your Changes:

- In the Source Control view, click on the commit message box.
- Write a descriptive message summarizing the changes you made.
- Click the checkbox next to staged files to confirm what's included in the commit.
- Click the checkmark icon (or press Ctrl+Enter) to commit your changes.

Push Changes to GitHub:

- Make sure you have a GitHub account and a remote repository created for your project.
- In the Source Control view, click on the **Publish to GitHub** button (arrow icon pointing upwards). This opens a GitHub login flow if not already connected.
- Choose the appropriate remote repository and branch where you want to push your changes.
- Click "Push" to upload your local commits to the remote repository on GitHub.