Installation and Navigation of Visual Studio Code (VS Code)

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.

* Make sure machine is running Windows 11.
* Download Visual Studio code from the official website.
* Make sure the downloader finishes and install on your machine.

1. 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.
   * Change the theme, font and size of the environment in settings.
   * Install extensions that you will you use, for example python and live server.
   * Enable Auto-save from settings
2. 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.
   * Activity Bar: Quick access to different views (Explorer, Search, Source Control, etc.).
   * Side Bar: Displays contextual tools and information related to the selected view.
   * Editor Group: Central area for writing and editing code, supporting multiple tabs and split views.
   * Status Bar: Provides information about the current state of the workspace and editor, such as file details, Git status, and error counts.
3. 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.

The Command Palette in Visual Studio Code is a powerful feature that allows users to access and execute a wide range of commands and actions quickly without having to navigate through menus or remember keyboard shortcuts. It provides a convenient way to perform tasks related to editing, navigation, version control, debugging, and more.

* + Type Search to search for text within your project files.
  + Type Replace to find and replace text.
  + Type Git: Clone to clone a repository.

1. 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.

Extensions in Visual Studio Code play a crucial role in enhancing the functionality and customizing the editor to fit various development needs. They allow users to add features and tools that streamline the coding process, improve productivity, and support specific programming languages, frameworks, and workflow

* + - HTML Snippet
    - CSS Peek
    - Prettier
    - React Native Tools

1. 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?

Go to View then Terminal from the top menu bar to open the integrated termina

* + - The integrated terminal allows you to run commands, scripts, and other terminal tasks without leaving the VS Code environment. This seamless integration improves workflow efficiency.
    - The integrated terminal automatically opens in the root directory of your workspace, making it easier to run project-specific commands without navigating directories.
    - Switching between editing code and running terminal commands is quick and easy, with no need to switch windows or applications.

1. 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?

* Use the Explorer view to right-click and select options for creating new files or folders. Alternatively, create a new untitled file with options in the File menu and save it accordingly.
* Access files and folders through the File menu by selecting the appropriate options. Click on items in the Explorer view to open them directly. The Command Palette also provides quick access.
* Rename files and folders by right-clicking and choosing the rename option. Move items by dragging and dropping within the Explorer view or use cut and paste commands. Delete files and folders through right-click options.
* Navigate efficiently using the Explorer view for browsing, Quick Open for fast file access, and tabs for switching between open files. Use integrated tools like the terminal and sidebar views for enhanced navigation and task management.

1. 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.

Use the Settings UI accessed by clicking on the gear icon

* Changing the Theme: Modify "workbench.colorTheme" in settings to switch themes.
* Adjusting Font Size: Use "editor.fontSize" to change the size of text in the editor.
* Customizing Keybindings: Edit the "keybindings" array in settings to define custom keyboard shortcuts.

1. 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?

Debugging in Visual Studio Code (VS Code) involves setting up and executing a debugger to find and fix issues in your code.

* Variable Inspection: Hover over variables to see their current values or add them to the Watch panel.
* Call Stack: View the sequence of function calls that led to the current point in the code execution.
* Debug Console: Interact with your running code via a console interface for logging messages or executing commands.

1. 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.

Visual Studio Code (VS Code) seamlessly integrates with Git, providing tools to manage version control directly within the editor

* To initialize a repository in the terminal, use the command ‘git init’
* To commit, use ‘git add .’ then ‘git commit -m’
* To push changes to the repositories use ‘git push’