Day 1 Reading:

1. VSCode:
    a. Getting Started with VSCode: https://code.visualstudio.com/learn
    b. Configuring For Typescript: https://code.visualstudio.com/docs/typescript/typescript-tutorial
2. Visual Studio
    a. Getting Started with Visual Studio: https://visualstudio.microsoft.com/vs/getting-started/ - Pick Up **Desktop development using .NET Core** -> **Step-by-step tutorial ->** https://learn.microsoft.com/en-us/dotnet/core/tutorials/with-visual-studio?pivots=dotnet-7-0
3. Git
    a. Learn about Git: https://www.freecodecamp.org/news/learn-the-basics-of-git-in-under-10-minutes-da548267cc91/
    b. More Git Tutorial: https://www.youtube.com/watch?v=8JJ101D3knE
    c. Learn about GitHub: https://docs.github.com/en/get-started/quickstart/hello-world

    a.

Practical Exercise 1:

Objective: Create a simple C# console application using Visual Studio, manage the project using Git, and host the repository on GitHub.

1. Set up the development environment:
    a. Install Visual Studio and VSCode
    b. Install Git
    c. Create a GitHub account
2. Create a new C# console application in Visual Studio:
    a. Create a new project, choose the "Console App (.NET Core)" template
    b. Name the project "HelloWorldApp"
3. Write a simple C# program:
    a. In the `Program.cs` file, modify the `Main` method to print "Hello, World!" to the console
4. Test the application:
    a. Run the application in Visual Studio to ensure it works correctly
5. Initialize a Git repository for the project:
    a. Open the terminal (command prompt or PowerShell) and navigate to the project folder
    b. Run `git init` to initialize a new Git repository

6. Configure Git user information:
    a. Run `git config user.name "Your Name"`
    b. Run `git config user.email "youremail@example.com"`
7. Commit the initial version of the project to the Git repository:
    a. Run `git add .` to stage all files for commit
    b. Run `git commit -m "Initial commit"` to commit the staged files

8. Create a new GitHub repository:
    a. Log in to your GitHub account
    b. Create a new repository named "HelloWorldApp"
9. Link the local Git repository to the GitHub repository:
    a. Run `git remote add origin https://github.com/yourusername/HelloWorldApp.git` (replace the URL with the one for your repository)
10. Push the local commits to the GitHub repository:
    a. Run `git push -u origin master`
11. Verify the project is available on GitHub:
    a. Check your GitHub repository to ensure the "HelloWorldApp" project files are present

Practical Exercise 2: Debugging with Visual Studio

Objective: Debug a simple C# console application in Visual Studio to understand the process of debugging.

1. Create a new C# console application in Visual Studio or use the "HelloWorldApp" from the previous exercise.
2. Add a new method called `Sum` that takes two integer arguments and returns their sum.
3. In the `Main` method, call the `Sum` method with two integers and print the result to the console.
4. Introduce a deliberate error by changing the addition operation in the `Sum` method to subtraction.
5. Set a breakpoint on the line with the erroneous operation in the `Sum` method.
6. Run the application in Debug mode.
7. When the breakpoint is hit, inspect the values of the two integer arguments.
8. Step through the code to see the incorrect result being returned.
9. Stop debugging, fix the error in the `Sum` method, and re-run the application to ensure it works correctly.

Practical Exercise 3: Debugging with VSCode

Objective: Debug a simple JavaScript application in VSCode to understand the process of debugging.

1. Create a new folder for a JavaScript project and open it in VSCode.
2. Create an `index.html` file with a basic HTML structure and a script tag to include a `main.js` file.
3. Create a `main.js` file in the same folder.
4. In `main.js`, write a simple function called `multiply` that takes two numbers and returns their product.
5. Call the `multiply` function with two numbers and log the result to the console.
6. Introduce a deliberate error by changing the multiplication operation in the `multiply` function to division.
7. Set a breakpoint on the line with the erroneous operation in the `multiply` function.
8. Run the application using the Debugger for Chrome extension in VSCode (install it if necessary).
9. When the breakpoint is hit, inspect the values of the two numbers.
10. Step through the code to see the incorrect result being returned.
11. Stop debugging, fix the error in the `multiply` function, and re-run the application to ensure it works correctly.

Practical Exercise 4: Working with Git – Pull Requests, Merge vs Rebase, Pull vs Push

Objective: Understand the concepts of pull requests, merging, rebasing, pulling, and pushing in Git.

1. Fork the "HelloWorldApp" repository on GitHub (created in the previous exercise) to your own account.
2. Clone the forked repository to your local machine.
3. Create a new branch called `feature/sum-multiplication`.
4. In the new branch, add a new method called `Multiply` that takes two integer arguments and returns their product.
5. Commit your changes to the new branch.
6. Push the new branch to your forked repository on GitHub.
7. Create a pull request from the `feature/sum-multiplication` branch to the `master` branch of the original repository.
8. In the original repository, merge the pull request using the "Create a merge commit" option.
9. In the forked repository, update the `master` branch with the latest changes from the original repository using `git pull`.
10. Create another branch called `feature/division` and add a new method called `Divide` that takes two integer arguments and returns their quotient.
11. Commit your changes to the new branch and push it to your forked repository on GitHub.

12. Create a pull request from the `feature/division` branch to the `master` branch of the original repository.
13. In the original repository, merge the pull request using the "Rebase and merge”