**Questions:**

**Introduction to GitHub:**

What is GitHub, and what are its primary functions and features? Explain how it supports collaborative software development.

* Github is a website for hosting repositories.
* Its main function is to allow seamless collaboration through sharing of source code and clear tracking of any changes committed to the source code.
* It supports collaborative development by allowing access to the source code for everyone working on the project and a clear history of any changes made to the source code.
* Github has features such as github gists which allows sharing of code snippets and github pages that enable hosting of web pages.

**Repositories on GitHub:**

What is a GitHub repository? Describe how to create a new repository and the essential elements that should be included in it.

* Github repository is a storage on github for your code that allows for collaboration, tracking changes and retrieval of the code.
* A new repository can be created as follows:

1. Log in to github.com
2. Click on the + icon on the top right of the page to display a menu
3. Click new repository on the menu to display a new page
4. On the new page input the name and description of the new repository in the relevant fields. You can chose whether to keep it public or private and whether to add a README file or note.
5. Click the Create Repository button and your repository has been created

**Version Control with Git:**

Explain the concept of version control in the context of Git. How does GitHub enhance version control for developers?

* Version control is a system that tracks changes made to a project.
* Github enhances version control by allowing for the changes to be stored and later referenced in the future if need arises. It also allows the changes to be reverted to a previous version.

**Branching and Merging in GitHub:**

What are branches in GitHub, and why are they important? Describe the process of creating a branch, making changes, and merging it back into the main branch.

* Branches is a feature that allow for working on different features without making changes to the main branch.
* This can be done as follows:

1. Create a branch using the command ‘git branch <name-of-branch>’
2. Move into the new branch using the command ‘git checkout <name-of-branch>’
3. Make the desired changes and add them to the staging area using the command ‘git add .’
4. Commit the changes using the command ‘git commit -m “message describing the changes”’
5. Move back into the main branch using the command ‘git checkout main’
6. Merge the changes into the main branch using the command ‘git merge <name-of-branch>’

**Pull Requests and Code Reviews:**

What is a pull request in GitHub, and how does it facilitate code reviews and collaboration? Outline the steps to create and review a pull request.

* A pull request is a request to merge changes from one branch into another.
* It allows for the changes to be reviewed and approved before being incorporated.
* Steps to create a pull request:
  1. Go to the repository on github
  2. Click the ‘pull requests’ tab
  3. Click the ‘New pull request’ button
  4. Select the branch with the changes and the branch to merge into
  5. Add a title and description
  6. Click the ‘Create pull request’ button
* Steps to review a pull request:
  1. Click the ‘pull requests’ tab
  2. Select the pull request to be reviewed
  3. Click the ‘files changed’ tab to see the changes made. If changes are accepted, click ‘review changes’ and select ‘approve’. If changes are not accepted click ‘review changes’ and select ‘request changes’
  4. Click the ‘merge pull request’ button to incorporate the changes

**GitHub Actions:**

Explain what GitHub Actions are and how they can be used to automate workflows. Provide an example of a simple CI/CD pipeline using GitHub Actions.

* Github Actions is a way to automate tasks such as building, testing, deploying, and releasing code from your gitHub repository.
* Can be used to automatically run tests, deploy applications and perform routine tasks.

**Introduction to Visual Studio:**

What is Visual Studio, and what are its key features? How does it differ from Visual Studio Code?

* Visual studio is an integrated development environment(IDE)
* Its key features include a code editor, a debugger, extensions, performance profilers and testing tools
* It differs from vs code since it is more resource-intensive and has a more comprehensive developer environment than vs code.

**Integrating GitHub with Visual Studio:**

Describe the steps to integrate a GitHub repository with Visual Studio. How does this integration enhance the development workflow?

**Debugging in Visual Studio:**

Explain the debugging tools available in Visual Studio. How can developers use these tools to identify and fix issues in their code?

**Collaborative Development using GitHub and Visual Studio:**

Discuss how GitHub and Visual Studio can be used together to support collaborative development. Provide a real-world example of a project that benefits from this integration.