1. **What is GitHub?**
    - **GitHub** is a cloud-based platform where developers can store, share, and collaborate on code.
    - **Primary Functions and Features**:
        - **Repositories**: Organize code, track changes, and allow collaboration.
        - **Branching**: Work on different versions of a project simultaneously.
        - **Pull Requests (PRs)**: Propose changes, review code, and collaborate.
        - **Issues**: Track tasks, enhancements, and bugs.
        - **GitHub Actions**: Automate workflows (e.g., CI/CD).
2. **GitHub Repository**:
    - A **repository (repo)** is where project files live.
    - To create a new repository:
        - Click “New repository.”
        - Provide a name, description, and choose visibility (public or private).
        - Initialize with a README (essential for project info).
        - Add a license (e.g., MIT License).
3. **Version Control with Git**:
    - **Git** tracks changes to files over time.
    - **Branches** allow parallel development.
    - **Merging** combines changes from one branch into another.
4. **Branching and Merging in GitHub**:
    - **Branches**:
        - Isolate work (e.g., features, fixes).
        - Create with `git checkout -b branch-name`.
    - **Merging**:
        - Combine changes from one branch into another.
        - Resolve conflicts if needed.
5. **Pull Requests and Code Reviews**:
    - **Pull Request (PR)**:
        - Proposes changes from one branch to another.
        - Facilitates code reviews and collaboration.
    - **Steps**:
        1. Create a branch.
        2. Make changes.
        3. Open a PR.
        4. Reviewers provide feedback.
        5. Merge after approval.
6. **GitHub Actions**:
    - Automate workflows (e.g., CI/CD pipelines).
    - Example: Automatically build and test code on every push.
7. **Introduction to Visual Studio**:

- **Visual Studio**:
  - Integrated development environment (IDE).
  - Key features: code editing, debugging, testing, project management.
  - Differs from **Visual Studio Code** (lightweight code editor).

8. **Integrating GitHub with Visual Studio**:
   - Steps:
     1. Install Visual Studio.
     2. Connect to GitHub.
     3. Clone a repository.
     4. Work locally and push changes.
9. **Debugging in Visual Studio**:
   - Tools:
     - **Breakpoints**: Pause execution for inspection.
     - **Watch window**: Inspect variables.
     - **Immediate window**: Execute code during debugging.
10. **Collaborative Development using GitHub and Visual Studio**:
    - GitHub and Visual Studio together:
      - Enable seamless collaboration.
      - Example: Developing a web app, reviewing PRs, automating workflows.
