GitHub-it’s a web based version control and interactive development software for collaboration by developers, built around Git version control system allowing multiple developers to work on projects simultaneously.

Functions of github:

* Version control – GitHub leverages Git version control system to track changes to code, manage versions and maintain complete history of modifications.
* Repositories- this are storage spaces for projects they act as folders to store different projects
* Pull requests- this is where developers notify each other of changes made to a project they are working on.
* Issues and project management- GitHub issues are used to track bugs, enhancements, tasks and other project related items.
* Continuous integration and deployment- GitHub integrates various tools enabling automated testing and deployment.
* Documentation – repos have form in the form of readme file which can be used to describe code and also documentation of the project being worked on
* Branching and merging – this allows developers to create separate lines of development within a repository.

It supports collaborative software development through:

* Centralized code repository
* Version control and history
* Branching and merging
* Pull requests and code review
* Automated work flow and collaborate
* Issue tracking and management

Q2:

Git repository is a location where project code, files and documentation on GitHub are stored.it allows developers track changes, manage versions and collaborate with others.

To create a repository;

Login to GitHub, clock on create new repository in the top right corner of the GitHub interface,

Enter new repo name, description, indicate whether private or public, add a readme file which is optional and a license which is also optional and click on create repository.

Q3:

Version control refers to the ability to track changes to files over time so that you can recall specific versions later.

Git enables different developers to work on projects simultaneously without interfering with each other`s work.

**B**ranching is creating separate lines of development within a repository, allowing developers to work on different features in isolation from the main codebase.

Merging is integrating changes from one branch into another.

Q4:

Branches are separate lines of development.

Importance.

* Branches allows developers to isolate their work.
* Branches allow parallel development; Multiple developers can work on different features or fixes simultaneously.
* Code Quality-Branching allows code reviews which in turn makes the code better as different people can access the branch and review.
* Branching also allows experimentation of new ideas, if the experiment is successful the merging can take place.

**Q5:**

**Creating a new branch:**

Login to GitHub, clock on create new repository in the top right corner of the GitHub interface,

Enter new repo name, description, indicate whether private or public, add a readme file which is optional and a license which is also optional and click on create repository

**To make changes to a branch**;

Git add .

Git commit –m “this is making changes to a branch”

Git push origin main

**Merging a Branch into the Main Branch:**

Create a pull request in GitHub from the featured branch to the main branch, after reviews the pull request is merged to the main branch using this code into the command line interface.

git checkout main

git merge feature-branch

A pull request is a process of proposing changes to a codebase.

To create pull request; push your branch to GitHub, go to your repository, click on the *pull request* tab, click on the *new pull request* button then select branches to merge *add title and description* then finish by clicking a *pull request button*.

Code Reviews involves other team members looking at the proposed changes to ensure code quality, consistency and functionality.

Q6:

Pull request refers to a way of developers to notify team members that they have completed a feature or debugging ready for review.

Click on the pull requests tab

Click the new pull request button

Select the base branch(main)

Add title and description for the pull request explaining the purpose

Click on the create pull request button to effect the request.

Q7:

GitHub Actions is a feature in GitHub that allows developers to automate workflows directly into their repositories.

Create a .github/workflows Directory

Create a file named ci.yml inside .github/workflows

Define the workflow in ci.yml

Q8:

Vsual studio is a comprehensive intergrated environment developed to help developers build programs, website and other web services on the internet.

Key features include;

Powerful debugging

Code editing features

Integrated source control

Collaboration tools

Comprehensive testing

Extensions and customization

Multi-platform development

Q9:

After installation of git and github then sign in into github from visual studio by;

Going into view, Team explorer, click on connect, on manage connections click on connect to github.

Enhancing Development workflow;

Seamless version control

Integrated code reviews

Collaboration

Productivity tools

Q10:

Debugging tools include breakpoint, watches, call stacks, locals, immediate windows.

How to use this tools to identify and fix issues include;

Set breakpoints – by setting this you identify where the issues might be and debug between the points.

Run the debugger the code will run up to the breakpoints.

Q11:

Github and visual studio can be seamlessly integrated to support collaborative environment enhancing the productivity and efficiency of software development teams.

* Sign in Github from visual studio
* Clone repository
* Create a new repository

Collaborative features include;

* Branching and merging
* Version control
* Pull requests and code reviews
* Issue tracking
* Code reviews
* Continuous integration

*Reference :*

*Chatgpt*

*Google.com*