ASSIGNMENT 2

GIT AND GITHUB QUESTIONS

1. Describe the steps required to install Git on a Windows machine. What key options should you pay attention to during installation, and why?

Visit the official Git for Windows website and choose the appropriate installer for your system, 32-bit or 64-bit, and run the installer.

Once the download is complete, choose installation options, finish the installation process and verify installation.

Choosing the Environment PATH – Adding Git to the PATH simplifies the process of using Git commands from any location.

Choosing the Git Bash – Using Git Bash as your default terminal can provide an easier experience.

1. Explain the purpose of configuring your username and email in Git. How does this configuration affect your Git workflow?

In Git, your username and email address are used to identify you as the author of commits.

In collaborative projects, your username and email address help distinguish your contributions from those of other team members.

Your username and email address can also be used to associate commits with specific issues or pull requests.

1. What is an SSH key, and why is it recommended to connect Git to GitHub using SSH? Provide a step-by-step guide for generating and adding an SSH key to GitHub.

An SSH key is a pair of keys, a public key and a private key, used for secure authentication. When you connect to GitHub using SSH, you’re providing GitHub with your public key. GitHub then uses this public key to verify your identity when you try to connect using your private key.

Compared to using a password, SSH keys provide a much more secure way to use GitHub.

Once you have set up SSH keys, you can connect to GitHub without having to enter your password each time.

STEPS TO GNERATE AND ADD AN SSH KEY TO GITHUB

Generate a key pair

Locate the public key

Copy the public key

Add the public key to GitHub

Test the connection

1. Provide the Git commands for the following tasks and explain what each command does:
2. Initialize a new Git repository.

git init – Initializes the working directory where you are running the command.

1. Clone an existing repository.

git clone <repository url> - Creates a local copy of the specified repository on your machine and initializes a new Git repository in the current directory.

1. Add all modified files to the staging area.

git add . – Stages all modified files within the current directory to the staging area and prepares files for commit.

1. Commit the changes with a message.

git commit -m “Your commit message” – Creates a new commit in your Git repository, records the changes made since the last commit and adds the commit message to make it easier to understand the purpose of the changes.

1. Push the changes to the main branch on GitHub.

git push origin main – Sends changes to remote repository and updates the main branch on the remote repository with your local changes.

1. After setting up Git and GitHub, how can you verify that your local Git setup is properly connected to GitHub? What is the expected output?

You can use the command to check configured username or email:

git config –global user.name

git config –global user.email

This command will display your configured username or email

PYTHON NAVIGATOR QUESTIONS

1. Explain the concept of variables and data types in Python. Provide an example in Python where different data types (integer, string, and Boolean) are used.

Variables are like containers that store data in a Python program and data types define the kind of data a variable can hold.

Example:

age = 20 #Integer

name = “Abigail” #String

is\_student = true #Boolean

print(age)

print(name)

print(is\_student)

1. What is control flow in Python? Write a Python script using if, elif, and else statements to check if a number is positive, negative, or zero.

Control flow refers to the order in which statements are executed in a program.

number = int(input(“Enter a number: “))

if number > 0:

print(“positive number”)

elif number < 0:

print(“negative number”)

else:

print(“zero”)

1. Differentiate between for loops and while loops in Python. Provide examples of each where a list of numbers is iterated over, and only even

numbers are printed.

“for loop” directly iterates over a sequence of values and a number of iterations is typically known in advance, and “while loop” continues to execute as long as a specified condition remains true and the number of iterations is not always known beforehand.

For loop:

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for number in numbers

if number %2 == 0:

print(number)

While loop:

numbers = [1, ,2, 3, 4, 5, ,6 ,7, 8, 9, 10]

index = 0

while index < len(numbers):

number = numbers[index]

if number %2 == 0:

print(number)

index +=1

1. Define what a function is in Python and explain its importance. Write a Python function that takes two arguments (a and b) and returns their sum.

A function in Python is a block of reusable ode that performs a specific task. It helps to organize your code, make it more modular, improve readability and allow reusability thus avoiding code duplication.

def calculate\_sum(x, y) :

“ “ “Calculates the sum of two numbers.

Args:

x: First number

y: Second Number

Returns:

The sum of x and y

“ “ “

return x + y

result = calculate\_sum(4, 5)

print(result)

1. Compare lists and dictionaries in Python. How would you use a list and a dictionary to store the names and ages of three people? Provide a

Python code example.

Lists and dictionaries are ways to store collections of data. Lists are ordered and indexed by numbers while dictionaries store data in key-value pairs.

List

names = [“Abigail”, “Cynthia”, “Samuel”, “Nella”]

ages = [20, 25, 16, 1]

dictionary

people = {

“Abigail” = 20,

“Cynthia” = 25,

“Samuel” = 16,

“Nella” = 1

}