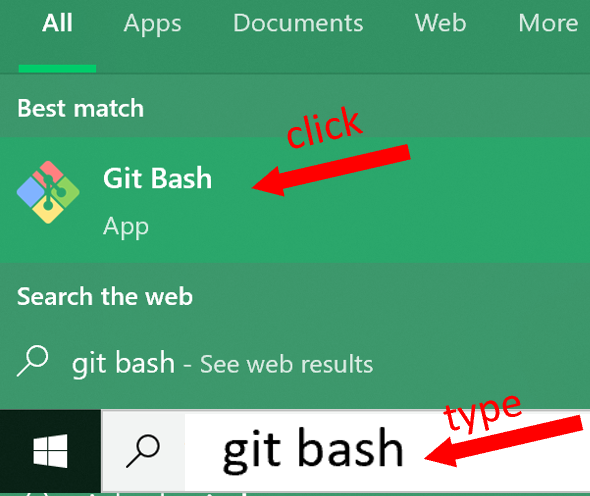
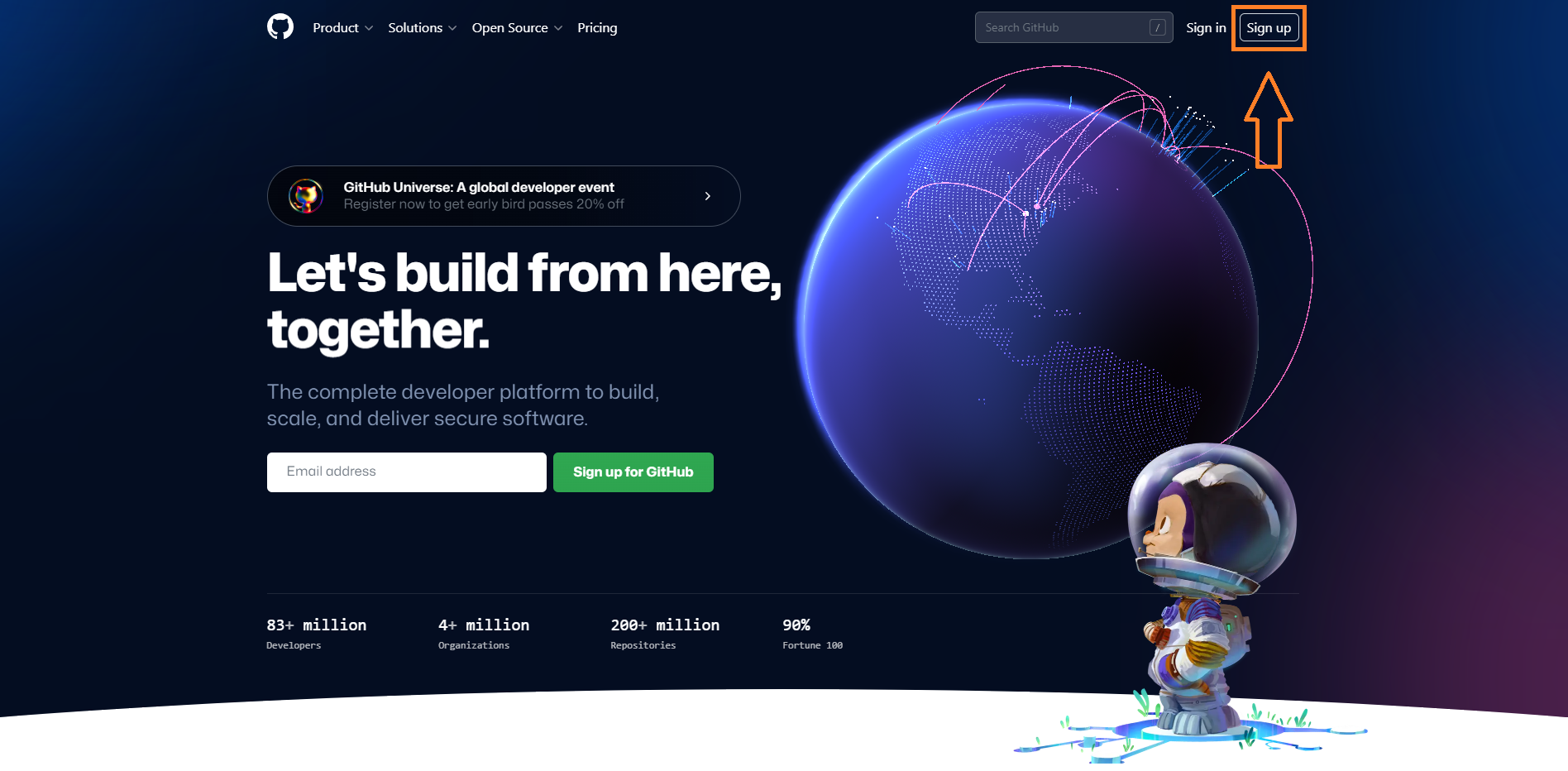
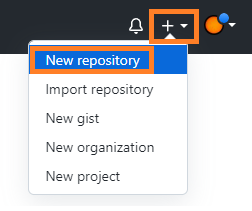
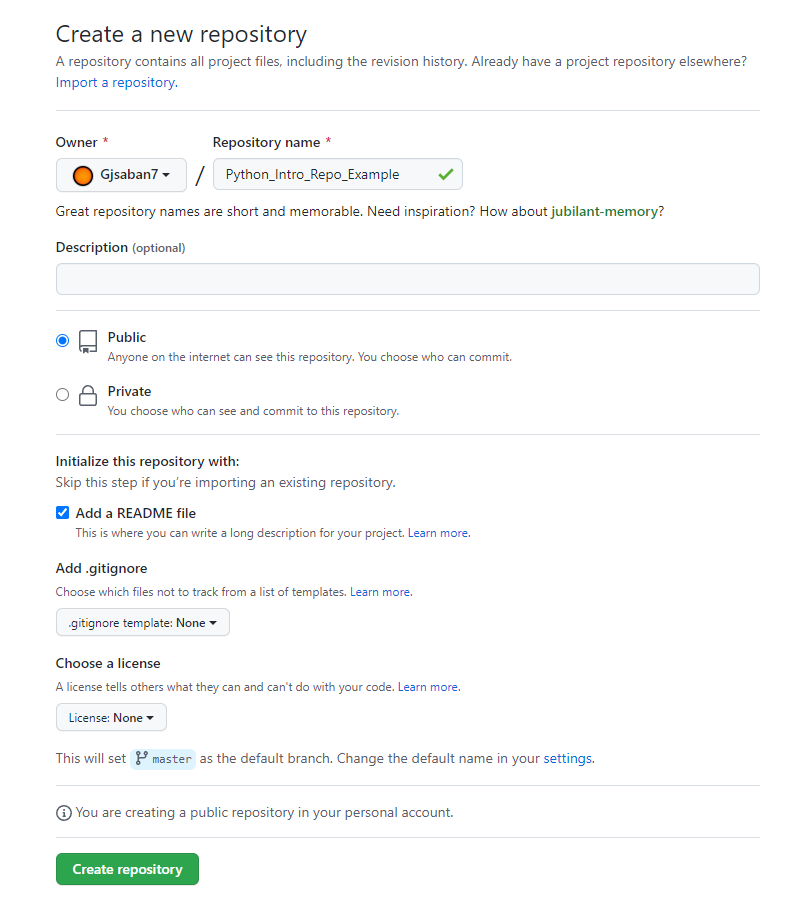
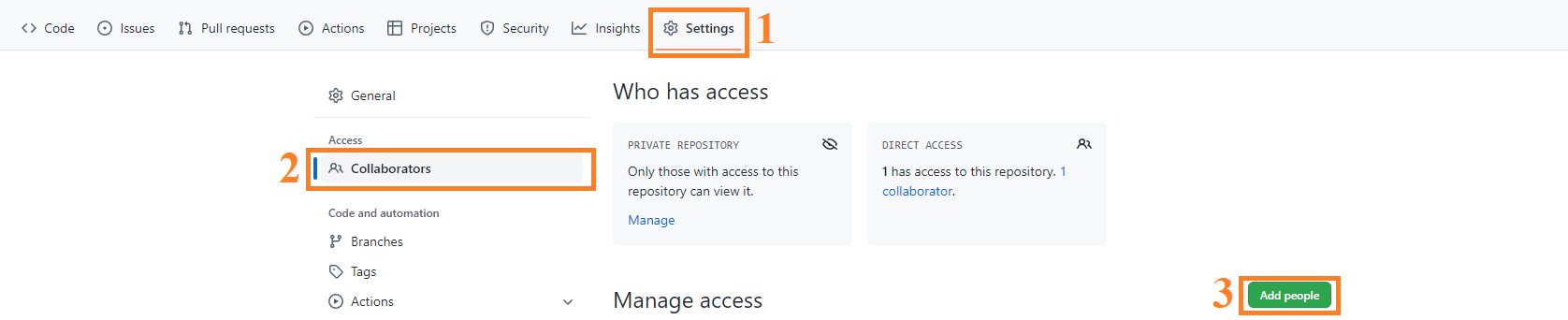
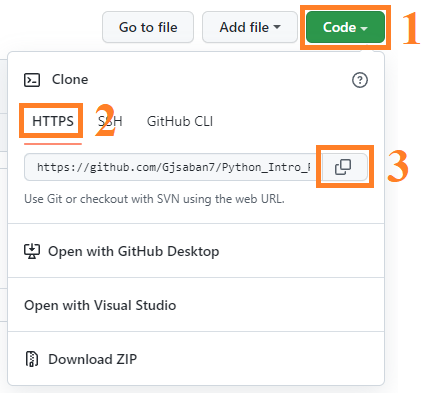
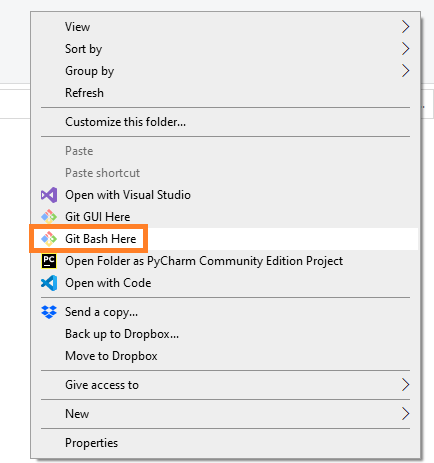
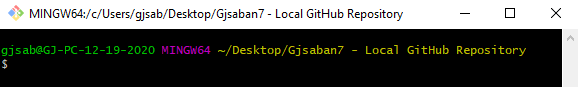
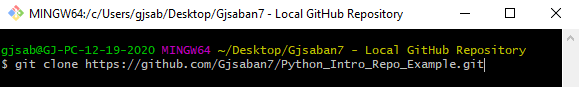
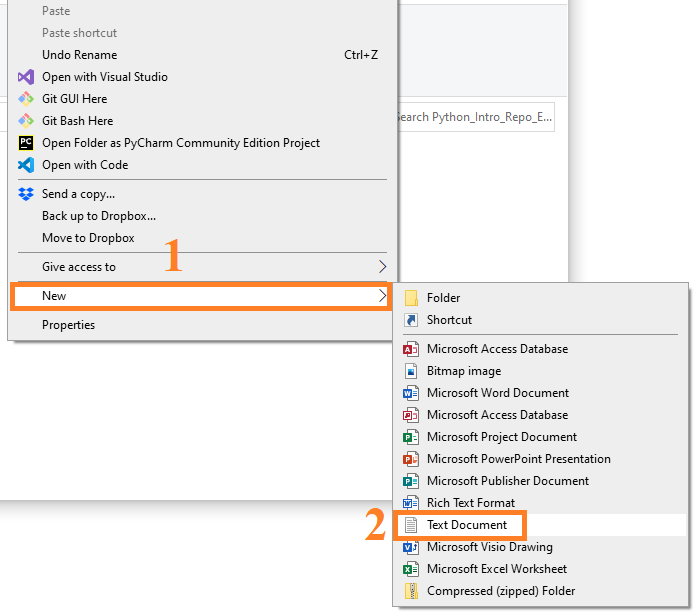
**GitHub Basics Tutorial**

(Updated Oct. 7, 2023)

* Git is a free and open-source software that offers a version control system that allows programmers to store, manage, and keep track of a source code history. Furthermore, GitHub is a cloud-based hosting service that allows the management of Git repositories online, making team collaborations and sharing of code easier. This system is analogous to Google Docs’ version history feature, where documents can be shared among users, and the document content can be reverted to a previous point if a rollback is necessary.
* To begin, check if Git is already installed in your local machine by opening a command prompt terminal and executing “git --version”. The output should look similar to the following:
  + 
  + If the version you have is not up to date, you should update to the most recent version by following the next step. Notwithstanding, older versions are fine for the minimal task that we will be using git for.
* Otherwise, if Git is not installed, navigate to [git-scm.com](https://git-scm.com/downloads) to download the latest version compatible with your operating system. For example, if on a Windows machine, select “Windows” and download the “64-bit Git for Windows Setup”. The new UI interface on the website might be slightly different.
  + A computer monitor with text and images

    Description automatically generated
  + 
* Upon successful installation, Git Bash (a command line emulation for GitHub commands) should now be installed and can be opened via the start menu search bar.
  + 
  + 
* Next, register or login into a [GitHub](https://github.com/) account to create a remote repository in the cloud
  + 
* Create a repository by selecting the “+” symbol located in the top right of the top navigation bar. Call your repo using your course name; for example, “GDS111-GSABAN”.
  + 
* When configuring the repository parameters, provide a repository name, select the “Public” radio button option, and enable the “Add a README file” checkbox. Leave the other fields as the default. For example:
  + 
  + Optionally, the repository can be made private to secure the codebase, but collaborators have to be manually shared in order for outsiders to view the repository. This would be accomplished by navigating to the “Settings” tab and “Collaborators” section, and searching for the other GitHub user’s username to allow them access to the repository.
    - 
* Clone a repository using the HTTPS protocol to create a copy and reference of the codebase stored in the cloud into your local system.
  + First, create a folder in your desktop (e.g. labeled “Local GitHub Repository”) to encapsulate the local repository to be cloned.
  + Second, return to the recently created repository and create a clone via selecting the “Code” button and copying the git URL. For example:
    - 
  + Next, within the created folder in the desktop, right-click to open the context menu and select “Git Bash Here”. For instance:
    - 
  + The resulting effect is a Git Bash application is opened and pointed to the folder’s relative path directory. For example:
    - 
  + Lastly, input the following command into Git Bash: “git clone [URL]”. The URL should be pasted into the ‘[URL]’ placeholder which was copied to the clipboard from an earlier step. For example:
    - 
    - *Note: To paste content from the clipboard, right-click and paste within Git Bash if the control+v keyboard shortcut does not work.*
    - A screenshot of a computer

      Description automatically generated
    - A screenshot of a computer screen

      Description automatically generated
* To demonstrate the basic features of Git, make some changes to the local repository and synchronize with the remote repository at GitHub.
  + First, create a file within the cloned repository by right-clicking to open the context menu, creating a new “Text Document”, and changing the file name to test.txt. For example:
    - 
    - A screenshot of a computer

      Description automatically generated
  + Afterward, open the file using any preferred editor (e.g. Notepad or Notepad++) and print a simple “Hello World” message. For instance:
    - A screenshot of a computer

      Description automatically generated
  + Next, save the changes and close the editor. At this point, close any previously opened Git Bash applications. Afterward, right-click within the file explorer to open the context menu and select “Git Bash Here” again to open a Git Bash application pointing to the correct path directory.
    - A screenshot of a computer

      Description automatically generated
    - In this demonstration, only one file was created within the local repository. To check this, type “git status”. The output would look similar to the following:
      * A computer screen with text on it

        Description automatically generated
    - To add all of the changes made locally to a staging area, execute the command “git add **.**”. No output should be shown. To check if the files are now tracked in the staging area, type “git status” again. The added file should be shown in green, such as:
      * A screen shot of a computer

        Description automatically generated
      * A screen shot of a computer

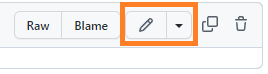
        Description automatically generated
    - Once all files intended to be pushed to the remote repository are added to the staging area, create a commit. Manually type “git commit -m ‘[message]’” (do not use copy-and-paste) to create a snapshot of the local repository. For example:
      * A screen shot of a computer

        Description automatically generated
    - If the “git status” command was executed again, the output message indicates that there is at least 1 commit ready to be pushed to the remote repository:
      * A screen shot of a computer

        Description automatically generated
    - Finally, enter “git push origin master” to push the local commits to the remote repository’s master branch.
      * *Note: Due to recent changes in GitHub, the default branch of “master” may have been relabeled to “main” for newer users. If “git push origin master” failed, try “git push origin main” instead.*
      * A screen shot of a computer

        Description automatically generated
    - If prompted, enter your GitHub credentials to authenticate Git Bash to interact with your GitHub repository.
    - At this point, return to the browser with the GitHub repository opened and refresh the page. Notice that the added file now exists in the remote repository. For instance:
    - A screenshot of a computer

      Description automatically generated
* Suppose a collaborator pushed their own changes to the remote repository or you pushed your own changes to the remote repository from a different system. How would you retrieve those changes to interact with locally? To demonstrate this scenario, make a change to the recent file using GitHub’s editor.
  + First, click select the file within the GitHub repository:
    - A screenshot of a computer

      Description automatically generated
  + Next, select the pencil icon or press the “e” key on the keyboard to edit the file in the cloud. For instance:
    - 
  + On a new line, create another simple print statement such as:
    - A screenshot of a chat

      Description automatically generated
  + Keep the default settings and select “Commit changes…”
    - A screenshot of a chat

      Description automatically generated
* To synchronize changes made in the remote repository with the local repository, pull the changes locally.
  + Return to the Git Bash application. Alternatively, if the application was closed, reopen the local repository (e.g. labeled “Local GitHub Repository”), right-click within the file explorer to open the context menu, and select “Git Bash Here” to open a Git Bash application pointing to the correct path directory.
  + Type and enter “git pull origin master” (or “git pull origin main”) (or “git pull” for short) to reflect any changes from the remote cloud repository’s master branch to the local repository. The result would look like the following:
    - A computer screen with text and numbers

      Description automatically generated
  + Close the Git Bash application and within the file explorer, open the text file using a preferred editor. Notice that the changes made in the cloud have been successfully pulled locally:
    - A screenshot of a computer

      Description automatically generated
* To learn more Git commands and features, consider exploring documentation such as [GitHub Docs](https://docs.github.com/en).

**Your Turn! Complete the tasks below, place your name and date, and submit the completed assignment to Canvas.**

* Paste below your github.com UI interface below. The remote cloud repository should look like the example:

|  |
| --- |
| Paste yours here … |

* Paste below your Git Bash console. It should show the status of your local repository or whatever was the last command you issued.

|  |
| --- |
| Paste yours here … |

* Finally, paste below your local repository folder’s content. The files in your folder may be different from the example.

|  |
| --- |
| Paste yours here … |

[](http://images.google.com/imgres?imgurl=www.skyscript.co.uk/im/trophy.jpg&imgrefurl=http://www.skyscript.co.uk/im/&h=214&w=180&sz=6&tbnid=ECCiP8U-7NsJ:&tbnh=99&tbnw=84&prev=/images?q=trophy&svnum=10&hl=en&lr=&ie=UTF-8&oe=UTF-8&sa=G)Congratulations! You’ve just learned how to use a source code management (SCM) program such as git/Git Bash/github.

1. Write your **name** here: \_\_\_\_Robert White\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Put **date** here: \_\_\_\_\_\_10-10-23\_\_\_\_\_\_\_\_