

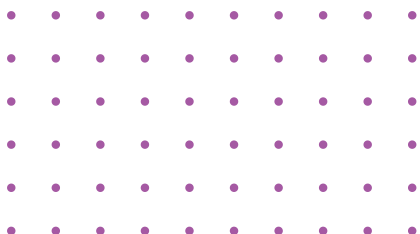
EXAMINATION

**INTRO TO COMPUTER
SCIENCE**

SUMMER 2024



LINNAEUS
UNIVERSITY
SUMMER 2024



EXAMINATION PREP

To complete the examination, you need:

- **to have created a GitHub account.** This will be your personal account, not connected to the school. This means that you can choose whichever email/username you wish. For now, set it to public, but after the course, you are free to do whatever you want with it!
- **to have downloaded Git on your computer.**
- **to have download and install Git Bash**
- **created a Replit account.** This will be your personal account, not connected to the school.

Read through the examination before you start - there is more software you need to download - however, during some of the installations, you are directed to take screen shots, that are to be handed in as part of the examination.

If you do not know how to do a screen shot, google/ask Chat GPT how to do it on your operating system.

SET UP

Create a repository on your GitHub account. Name it:

`yourStudentID_Summer24_IntroToCS`

for example, with my student ID it would look like this:

`pm222hs_Summer24_IntroToCS`

In this repository, **include a README where you introduce yourself as a programmer. This does not need to be lengthy!**

A README file is usually a document that provides information about a project. Recently, there has been a trend of using README files as a personal profile or "About Me" section on GitHub. This allows users to showcase their interests and skills in a more personalized way. [Read more here.](#)

You could, for example, write if you're just starting to learn programming, or if you've dabbled in coding before. You could specify if you have a special interest (web development, front end, back end, JavaScript, python, etc).



HOW TO HAND IN

HOW TO HAND IN

A detailed guide on how to use git to hand in your work will be provided on the course website on the examination page. It is a PDF called “**Examination Help - How to use Git for Hand-ins**”, and whenever you need to use it, it will be referenced in this document.

For now:

- **CREATE 4 FOLDERS in the repository**
“yourStudentID_Summer24_IntroToCS on with the GitHub GUI on the GitHub website.
- **Name them as follows:**
 - yourStudentID_Summer24_IntroToCS_Week1
 - yourStudentID_Summer24_IntroToCS_Week2
 - yourStudentID_Summer24_IntroToCS_Week3
 - yourStudentID_Summer24_IntroToCS_Week4

All your hand-ins for each week should be committed to that specific weeks’ folder - for example, Week 1 assignments into “yourStudentID_Summer24_IntroToCS_Week1”.

- **Create a folder on your computer for each assignment for each of the weeks. This WILL make it easier to hand in your work when using Git.**
 - week1_assignment1_markdown
 - week1_assignment2_break_the_ice
 - week2_assignment1_following_tutorials
 - week2_assignment2_download_nodejs_and_npm
 - week2_assignment3_navigation_and_file_operations
 - week3_assignment1_hello_world
 - week3_assignment2_variables_datatypes_and_ifstatements
- **For Week 3 assignment 3 you will use a different method of adding your work to GitHub. Refer to “Initialize a Local Git Repository and Push to GitHub” in the “Examination Help - How to use Git for Hand-ins” PDF.**

WEEK 1

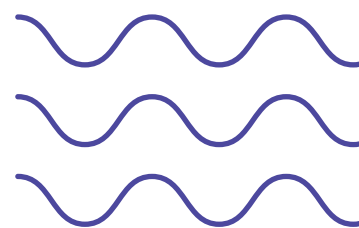
ASSIGNMENTS

ASSIGNMENT 1 - MARKDOWN DOCUMENT

- Write a Markdown document. This document will include questions for self-reflection, as well as some questions from the topics covered in this course. It will also include specific formatting requirements.
- Just 1-2 sentences per question is enough—the goal is to get you thinking, not writing essays! 😊 No need to feel pressured to write more than that.

MARKDOWN DOCUMENT - FORMATTING REQUIREMENTS

- Headings
 - At least two levels of headings (e.g., # for main headings and ## for subheadings) to organize the content.
- Code Blocks
 - Demonstrate how to create a code block to display code snippets. You can write anything you want in this code block.
- Lists
 - Include an ordered list with at least 2 items.
 - Include an unordered list with at least 2 items.
- Links
 - Create a hyperlink to a website of your choice.
- Emphasis and Bold
 - Use italics and bold formatting to highlight specific text within the document.
- Emoji
 - Add an emoji of your choice! 🚀
- Table
 - Create a simple table. 2x2 row and column is enough, but feel free to add more rows/columns if you want! You can write anything you want in the table.



MARKDOWN DOCUMENT - QUESTIONS

- What is your previous experience with computers, computer science and code? For example, have you never really had an interest before? Are you a gamer who has modded games? Have you tried coding a little?
- What made you interested in coding and decide to go to school to learn programming?
- What technology did you grow up with? Did you experience the world before Google and smartphones? If not, how do you think finishing it might have been different from navigating daily life without instant access to services, information, and communication?
- How does technology impact you positively, as well as negatively?
- Have you ever thought about how your personal data is used by companies? How do you feel about the way big data shapes your online experiences?
- Have you tried using ChatGPT or other generative AI tools in your work or hobbies? How did they help or change the way you approach tasks?
- Do you see code as logic or communication? What might be the benefit of each perspective?
- What mental skills listed in “Programming Mentality” do you find challenging? Write one actionable way to improve in that area!
- What did you expect from this course?
- Did you learn what you thought you would during this course? If not, what would you have wanted to be different?
- What did you enjoy most about this course, and what would you have liked to see more of?
- Check Your Disk Type - Use your computer’s disk management tool (e.g., Disk Management on Windows or Disk Utility on macOS) to check your disk type (HDD or SSD) and write it in the document (tip is to use the table for this!)
- Look up how to find your computer’s specifications. What operating system is it running on? What type of processor do you have?
- **Optional:** Write the non-sensitive computer specifications of your machine in your Markdown document. Processor, RAM, OS system version, and System Type are safe to share - they cannot be used to identify your machine or derive sensitive information that could be used maliciously. Write them out on a table

PLEASE DO NOT GIVE OUT ANY SENSITIVE INFORMATION. Here is a breakdown of what could be considered sensitive when sharing device specifications. Always be cautious with sharing details that could uniquely identify or compromise your device’s security.

- **SOFTWARE- Safe to Share:** OS Version, Build Number, System Type (32-bit/64-bit)
Avoid Sharing: Device Name, Product ID, Product Key.
- **HARDWARE - Safe to Share:** Device Name, Processor, Installed RAM, System Type, Pen and Touch. **Avoid Sharing:** Device ID, Product ID

HAND IN - CONTENTS

- **Folder Name:** week1_assignment1_markdown
- Inside the Folder: Add the Markdown document that addresses all the provided questions and meets all the necessary requirements.

All your hand-ins for the assignments for Week 1 are to be committed to the folder **yourStudentID_Summer24_IntroToCS_Week1**

Refer to “Examination Help - How to use Git for Hand-ins” - “Using Git & Clone a Repo” if you don’t have a repo cloned, or “Add Files/Folders to Already Cloned Repo”

ASSIGNMENT 2 - BREAK THE ICE

- Ask a question on Slack related to IT / programming that you might be struggling with & **take a screenshot of it.**
- Ask ChatGPT a question related to IT / programming that you might be struggling with & **take a screenshot of it.**

HAND IN - CONTENTS

- **Folder Name: week1_assignment2_break_the_ice**
- Inside the Folder: 2 screenshots of you asking questions using Slack and ChatGPT

All your hand-ins for the assignments for Week 1 are to be committed to the folder **yourStudentID_Summer24_IntroToCS_Week1**

Refer to **“Examination Help - How to use Git for Hand-ins”** - “Using Git & Clone a Repo” if you don’t have a repo cloned, or “Add Files/Folders to Already Cloned Repo”

ASSIGNMENT 3 - HARDWARE / SOFTWARE

- Task Manager/Activity Monitor- Open Task Manager (Windows) or Activity Monitor (macOS). Locate the CPU usage. Observe which processes are consuming the most CPU resources. Also, check the current RAM usage. Note which applications are using the most memory.

WEEK 2

ASSIGNMENTS

ASSIGNMENT 1 - FOLLOWING TUTORIALS

- **DURING THIS ASSIGNMENT YOU WILL BE ASKED TO TAKE A SCREENSHOT OF YOUR WORK.**
- **Follow each of the tutorials** “Terminal” and “File Systems” (available on the course site under Week 2) and take a **screenshot of your screen as you follow each tutorial.**

HAND IN - CONTENTS

- **Folder Name: week2_assignment1_following_tutorials**
- Inside the Folder: 2 screenshots of you following the tutorials “Terminal” and “File Systems” (1 screenshot each)

All your hand-ins for the assignments for Week 2 are to be committed to the folder **yourStudentID_Summer24_IntroToCS_Week2**

Refer to “Examination Help - How to use Git for Hand-ins” - “Using Git & Clone a Repo” if you don’t have a repo cloned, or “Add Files/Folders to Already Cloned Repo”

ASSIGNMENT 2 - DOWNLOAD NODE.JS AND NPM

- **AT THE END OF THIS ASSIGNMENT YOU WILL BE ASKED TO TAKE A SCREENSHOT OF YOUR WORK.**
- **Install Node.js and npm from the official node.js website.**
 - npm should be included in the node.js installation. You'll see two versions of node.js available: LTS (Long-Term Support) and the current version. Choose the current version - look for: "Want new features sooner?" on the site, and you should find a version that looks something like this: "Node.js v22.6.0". That's the one you want.
 - (For most users, the LTS version is recommended for code that will be used in live applications. However, during school they will recommend the current version so you can have access to new features).
- **Verify the installation & take a screenshot of Git Bash:**
 - Open Git Bash and run:
 - `node -v` to check the Node.js version.
 - `npm -v` to check the npm version.
- **Screenshot Git Bash with both of these version outputs clearly visible.**

HAND IN - CONTENTS

- **Folder Name: week2_assignment2_download_nodejs_and_npm**
- Inside the Folder: 1 screenshot of your terminal with the outputs for the commands `node -v` to check the Node.js version and `npm -v` to check the npm version.

All your hand-ins for the assignments for Week 2 are to be committed to the folder **yourStudentID_Summer24_IntroToCS_Week2**

Refer to "Examination Help - How to use Git for Hand-ins" - "Using Git & Clone a Repo" if you don't have a repo cloned, or "Add Files/Folders to Already Cloned Repo"

ASSIGNMENT 3 - NAVIGATION AND FILE OPERATIONS

- **AT THE END OF THIS ASSIGNMENT YOU WILL BE ASKED TO TAKE A SCREENSHOT OF YOUR WORK.**
- **Open Git Bash and navigate to the desktop folder using Git Bash and text commands.**
- **Create a directory.** Use Git Bash to create a directory - google /ask Chat GPT how to find the command to create a directory using Git Bash.
- **Check that the directory is created.** Google/ask Chat GPT how to find the command to view the contents of the current working directory.
- **Create a file using GUI.** Using the graphical user interface (using your mouse/keyboard and double-clicking) navigate into your newly created directory and create a .txt file. Add some text to the file and save the .txt file.
- **View .txt file contents using Git Bash.**
 - Use Git Bash to navigate into the directory you just created.
 - Check that the .txt file you created is present - use the text command that lets you view the contents of the current working directory to verify that the .txt file is there.
- **Find the absolute file path of your .txt file.** Either google/ask Chat GPT how to do it in the terminal, or use your OS's GUI to look up the file path. The path should be from the root of your filesystem.
 - On Windows, the root directory is represented by the drive letter followed by a backslash. For example, C:\ is the root directory of the C: drive.
 - On macOS and other Unix-like systems, the root directory is represented by a single forward slash /.
- **Copy/write down the absolute file path. YOU WILL NAME YOUR SCREENSHOT AFTER THIS PATH.**
- **View the contents of your text file.** Use the command `cat yourfilename.txt` in Git Bash.
- **Delete the file** using Git Bash. Use the command `rm yourfilename.txt`
- **View the contents of the current working directory again** (the .txt file should be gone).
- **Navigate out.** Move out of the directory in the terminal, to your desktop folder.
- **Delete directory** - remove the directory you created earlier using the command `rmdir yourdirectoryname`
- **Use the text command in Git Bash to view the contents of the desktop folder.** The directory you created should be gone.
- **Take a screenshot of the terminal window, showing as much of these operations described above as possible (resize the text size if necessary, so it's readable, but shows as much as possible)!**

HAND IN - CONTENTS

- **Folder Name:** week2_assignment3_navigation_and_file_operations
- Inside the Folder: 1 screenshot of your terminal with the output of the operations described above.

All your hand-ins for the assignments for Week 2 are to be committed to the folder **yourStudentID_Summer24_IntroToCS_Week2**

Refer to **“Examination Help - How to use Git for Hand-ins”** - “Using Git & Clone a Repo” if you don't have a repo cloned, or “Add Files/Folders to Already Cloned Repo”

WEEK 3

ASSIGNMENTS

ASSIGNMENT 1 - HELLO, WORLD!

- Go to Replit, log in, and create a new Repl. Select the programming language you want to use (e.g., JavaScript).
- Use `console.log()` to write your first "Hello, World!"
- Click the "Run" button to execute your program.
- View the output
- **Screenshot your result.**

HAND IN - CONTENTS

- **Folder Name:** `week3_assignment1_hello_world`
- Inside the Folder: 1 screenshot of your first "Hello, World!"

All your hand-ins for the assignments for Week 3 are to be committed to the folder **yourStudentID_Summer24_IntroToCS_Week3**

Refer to **"Examination Help - How to use Git for Hand-ins"** - "Using Git & Clone a Repo" if you don't have a repo cloned, or "Add Files/Folders to Already Cloned Repo"

ASSIGNMENT 2 - VARIABLES, DATATYPES AND IF-STATEMENTS

1 - Declaring and assigning values

- Go to Replit, log in, and create a new Repl. Select the programming language you want to use (e.g., JavaScript).
- Declare a Variable in the code editor - create a variable in your code using a keyword such as `let`, `const`, or `var`.
- On a separate line of code, assign a value to the variable.
- Print or display the value of the variable using `console.log()`. For example:

```
let age = 18
console.log(age)
```

- Reassign the same variable, giving it a new value.
- Use `console.log()` to print the updated value of the variable, then run the code to print the value again.
- **Screenshot your code and output.**

2 - Datatypes and typeof

- Initialize 3 new variables - 1 boolean (true/false), 1 string, 1 number.
 - String is a collection of characters - essentially, any text.
 - “Hello” is a string, as is “ “ and “Hi, there, coders!!! :)”
 - If you want to combine a string with a variable in the console log(), this is how:

```
javascript

let isRaining = true;
let city = "New York";
let temperature = 75;

console.log("Is it raining? " + isRaining);
// Output: Is it raining? true

console.log("The weather in " + city +
           " is currently " + temperature + " degrees.");
// Output: The weather in New York is currently 75 degrees.

console.log("It's " + temperature + " degrees in " + city +
           ". Is it raining? " + isRaining);
// Output: It's 75 degrees in New York. Is it raining? true
```

- Use `console.log(variableName)` to print the value of each of the variables.
- **Screenshot your code and output.**
- Use the command `typeof variableName` on each of the variables inside a `console.log()` - between the brackets. This will output their data type.
- **Screenshot your code and output.**

```
let isRaining = true;
let city = "New York";
let temperature = 75;

console.log("Type of isRaining: " + typeof isRaining);
// Output: Type of isRaining: boolean

console.log("Type of city: " + typeof city);
// Output: Type of city: string

console.log("Type of temperature: " + typeof temperature);
// Output: Type of temperature: number
```

3 - If statements

- Write an if statement for each variable.
- Screenshot your code and output for each of the 3 if statements.

NOTE!

Using **one =** assigns a value to a variable. Using **two ==** checks if two different values are equal.

```
let favoriteColor = "blue";  
  
if (favoriteColor == "blue") {  
    console.log("Your favorite color is blue!");  
}
```

For easy if-statements, use:

- **Equality Operator ==** (compares two values to determine if they are equal)
- **Relational Operators ><** (compares two values)
 - Greater Than (>): Checks if the value on the left is greater than the value on the right.
 - Less Than (<): Checks if the value on the left is less than the value on the right.

```
javascript  
  
let isRaining = true;  
let city = "New York";  
let temperature = 75;  
  
// Check if it is raining  
if (isRaining == true) {  
    console.log("It is raining.");  
} else {  
    console.log("It is not raining.");  
}  
  
// Check if the city is "New York"  
if (city == "New York") {  
    console.log("The city is New York.");  
} else {  
    console.log("The city is not New York.");  
}  
  
// Check if the temperature is greater than 70  
if (temperature > 70) {  
    console.log("The temperature is above 70 degrees.");  
} else {  
    console.log("The temperature is 70 degrees or below.");  
}  
  
// Check if the temperature is less than 80  
if (temperature < 80) {  
    console.log("The temperature is below 80 degrees.");  
} else {  
    console.log("The temperature is 80 degrees or above.");  
}
```

HAND IN - CONTENTS

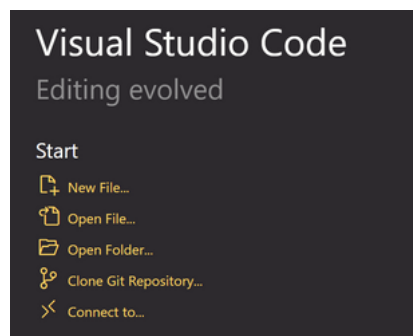
- **Folder Name: week3_assignment2_variables_datatypes_and_ifstatements**
- Inside the Folder:
 - 1 Declaring and assigning values - 1 screenshot
 - 2 Datatypes and typeof - 2 screenshots
 - 3 If-statements - 3 screenshots

All your hand-ins for the assignments for Week 3 are to be committed to the folder **yourStudentID_Summer24_IntroToCS_Week3**

Refer to “Examination Help - How to use Git for Hand-ins” - “Using Git & Clone a Repo” if you don’t have a repo cloned, or “Add Files/Folders to Already Cloned Repo”

ASSIGNMENT 3 - VSCODE & GIT

- Download VSCode.
- Download ANY extension in VSCode. One recommendation is Dracula - it's a theme that changes the colors of the UI. Feel free to choose any theme or extension you like! If you're unsure which one to start with, a theme is a great option.
- **Take a screenshot of VSCode and the extension installed on your computer.**
- Create a directory on your computer. Name it "Local_To_GitHub".
- Create a new project in VSCode. Open VSCode, and open the "Local_To_GitHub" folder when prompted **Open folder...** This will be the root folder of your project.

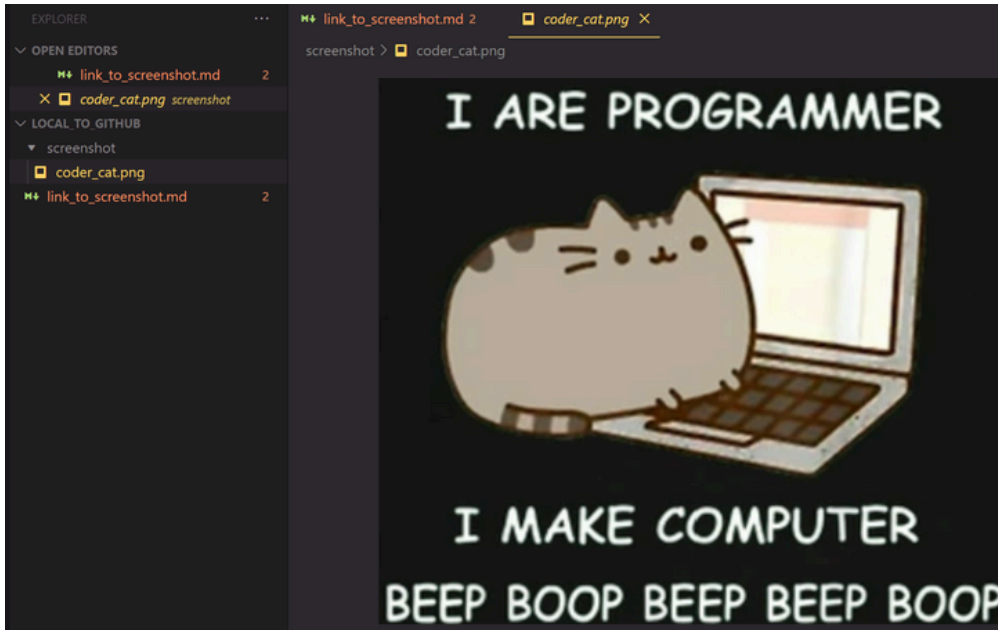


- Create a new folder named "screenshot" in your VSCode project "Local_To_GitHub". (Make sure to create it *inside* the "Local_To_GitHub" folder).
- Save the screenshot you've done of VSCode and the extension you installed inside the folder "screenshot".
- Create a .md file in the folder "Local_To_GitHub".
- Name your file "link_to_screenshot.md".
- Link to your screenshot (located in the folder "screenshot") *within* your Markdown file named "link_to_screenshot.md".
- Practice how to link to files using the relative path!

For this assignment you will use a different method of adding your work to GitHub.

Refer to "Initialize a Local Git Repository and Push to GitHub" in the "Examination Help - How to use Git for Hand-ins" PDF.

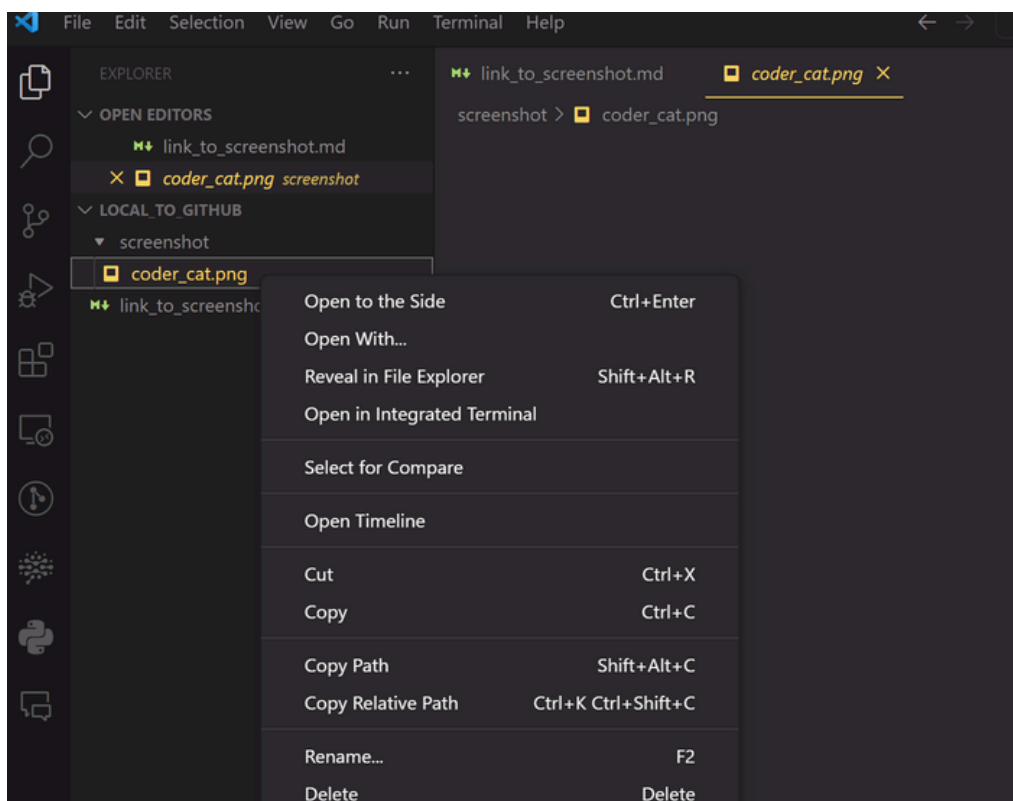
It should look something like this:



NOTE!

Right-clicking and choosing **“Copy”** will give you the **markdown syntax** for a link to your file, while **“Copy Relative Path”** WILL NOT.

Try **“Copy”**, **“Copy Relative Path”** and **“Copy Path”**, paste it in your .md file, and note the differences. Right-click on the .md file and chose **“Open Preview”**.



WEEK 4

ASSIGNMENTS

ASSIGNMENT 1 - MINI CALCULATOR

NOTE! More information is coming on these assignments.

Write a function with some type of math (you are welcome to copy from the tutorial!)
function - use it to perform a calculation.