

Guided Exploration M01: Programming Fundamentals and Tools

Points: 25 (See Rubric in Canvas)

Due Date: Due date listed in Canvas but some sections will be due as class participation before. Create Calendar reminders

- Final submission will be accepted up to 24 hours after the due date with a 10% penalty. Meaning if you turn it in at 12:01 am of the next day you will be deducted 10% of the total points from your score.
- If the assignment is more than 24 hours late, it will be a 0.

Submission: Upload files separately and do not upload a zip file.

- **Two Document Files as PDF or Word document:** This document with your answers in the highlighted areas unless otherwise stated and Technical Document
- **Two Java Files (. java files NOT CLASS FILES):** Code from 2.1 Guess Number and 2.2 Calculate Grades

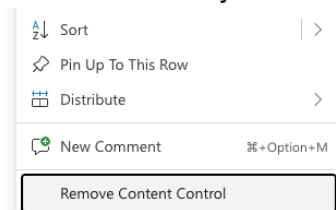
Objectives:

- Analyze and apply relevant theories, principles, and methods from computing, including algorithms, coding structures, and version control tools.
- Develop effective technical documentation to reflect on and solidify learning, with a focus on clarity and professional presentation.
- Apply version control and debugging practices using Eclipse and GitHub.

Effort: You are encouraged to collaborate to discuss concepts and explore writing code together. Remember

- Review the [lectures](#) and links in the lectures to information first.
- Follow [CS Academic Integrity and AI Policy - Harding](#)
- If you find you don't understand the information in the lecture reach out to **Deb, Elaine or Heriberto**

If you download this as a word document and you have any problems typing your answer in the tables you must click on 3 dots and remove content control.



[Part 0 Intro: Learning and Collaborating](#)

[0.1 Working on Teams](#)

[0.2 Learning How to Learn](#)

[0.3 Hello World](#)

[Part 1: Explore and Explain](#)

[Part 2 Analyze and Apply](#)

[2.1 Explore GuessNumber Code](#)

[2.2 Calculate Grades](#)

[3 Reflection](#)

Part 0 Learning and Collaborating Reflection

The reality of computer science is that there are so many languages, technologies and methodologies available and it is constantly evolving with new ones. So the goal is not just to understand the current technologies but developing skills to learn any new technology.

0.1 Working on Teams

In this course, we'll work together on in-class activities, labs, and discussions. Good collaboration isn't just about dividing work — it's about building understanding together. Read [10 Collaboration Skills Examples](#) and answer the following

1. Pick two of the 10 skills that you think are strengths you bring to the team and how you will bring those strengths to the team.

Problem solving and adaptability; With my background in military operations and high threat security management overseas, I have worked in a wide variety of environments, with an even broader spectrum of people. I am able to adapt to individual's work and/or communication style's within a team to find how to best work with them and produce results.

2 Pick two of the 10 skills you think are most difficult for you and something you can do to try and improve in that area this semester.

Trust building and empathy are things I could work on. In my past lines of work, trust came pretty naturally and quickly, due to the environment we were in. So building trust with people outside of life and death situations can sometimes be a challenge.

0.2 Learning How to Learn

Learning how to learn is an important component of this course as your career will be full of learning something new. One of my favorite websites is [Train Ugly - How To Get More Out Of Your Practice](#).

1 Listen to this 30 minute podcast [Desirable Difficulties - The Learner Lab](#) and answer the following.

Explain desirable difficulties and how you think it relates to the learning in this class and in your future career.

Challenges that you put in place for yourself to help increase the efficiency of whatever it is you are doing (studying, working out, practicing a skill, etc). I think this relates to this course and my future career in that it will be better for my learning to practice things in a variety of ways, accept that there is no ONE RIGHT WAY in coding, and that trying new things will help solidify what we are trying to learn, even if it doesn't work the first time around.

2 Read [Bluesky CEO Jay Graber warns: If you're a student, using AI means](#) where Bluesky CEO's Jay Graber warns that over-reliance on AI can impair critical thinking and long-term skill development and may contribute to "academic obsolescence" - declining ability to perform without AI

What skills do you risk losing if you depend on AI too early?

You risk losing critical thinking and problem solving skills, as well as disrupting your own learning process.

3 Summarize 3 Strategies you will use this semester to build skills, become a lifelong learner, and [Use AI Tools](#) in a way that supports your own thinking.

1. Study for understanding, not memorization. By studying to actually understand the content on a deeper level, instead of memorizing it for regurgitation later, I can use the knowledge gained in other scenarios or further down the line than if I did not understand it
2. Practice teaching the content, even if its just to a wall. I learned in my bootcamp that by helping other students when they got stuck, I was able to better solidify what I understood about the topic.
3. Using AI has been helpful for me in finding bugs; sometimes I will spend a bunch of time trying to find the extra { and AI will point it out immediately. Sometimes I will also run an idea by it, like if using a certain methods, conditionals, etc would be a good idea in a given context. As long as I can understand what I am inputting and getting back from it, I find it to be a solid learning and assistant tool. I plan to keep using it in the same way, and avoid using it as a crutch or replacement for actually learning the content myself.

0.3 Algorithms



"You can't just build an algorithm, put it on a shelf and decide whether you think it's good or bad completely in isolation. You have to think about how that algorithm actually integrates with the world that you're embedding in." Hannah Fry

Read [Chapter 1](#) from Hannah Fry, "Hello World, Being Human in the Age of Algorithms" and answer the following.

Answer the following in your own words.

1. Describe an algorithm.

A set of step by step instructions to accomplish a specific task. It could be a computer algorithm that calculates employee's pay, a cooking recipe "algorithm" that describes how to make a beef stew, or even a set of GPS instructions for how to get to work in the fastest manner

2. Summarize the four main categories of real world tasks that algorithms are doing.

- Prioritization: sorting a list based on the relevance or some measure of importance
- Classification: assigning a category to the subject, such as iOS' facial recognition in Photos, allowing users to search for photos that contain a certain person's face(photo is classified as having that person in it), or dating application's that classify users based on age, sexual preference, gender, etc.
- Association: deals with finding connections between things; This may involve taking those classifications on a dating app and comparing them to the classifications or preferences of another user to see if there is a match, or a link, between them.
- Filtering: involves showing content that is relevant, and removing content that might just be clutter. Some online stores may not show certain products if they don't ship to your location, or if the reviews are too low, etc.

3. Explain the two main approaches taken by algorithms. Include the pros and cons.

- Rule-based: Rule-based algorithms are algorithms written by humans, contain a set list of instructions, and are generally easier to read by other humans, since they were written by one. On the other hand, rule-based algorithms may struggle with more complex tasks, such as translating languages, detecting qualities in images, or searching text.
- Machine-Learning: based on giving data, an objective, and providing feedback on its progress towards meeting that goal. This allows the AI to tackle much more complex tasks that a human may not be able to write commands for, but this also means that a human may not have an easy time understanding what is happening behind the scenes in the algorithm or reading the “steps” that the AI took to reach its conclusion.

4. What can happen when we rely on machines to make decisions for us?

Becoming too reliant on machines to make decisions for us can lead us to complacency, a decline in critical thinking skills, and can also lead to huge errors, like the Medicaid algorithm that was mentioned in the text. Algorithms and machines can be used to assist us, but there should be checks and balances to make sure they are doing what they were intended to do.

Part 1: Explore and Explain

Purpose: Professional communication to demonstrate your learnings by connecting different ideas from lectures or provided resources.

- Clearly explain key ideas in your own words and include supporting evidence such as code examples, images, screenshots, or charts.

You will create a technical document to have as a resource for your projects, quizzes, future reference and resume.

- [Tech Doc CS1050](#): Download as word or create a copy of the google documentation.
 - Put in your git repository after you set up git and github.
 - Submit this as a word or pdf with your GE submission.
- Use headings to organize topics and create or update a table of contents to be able to quickly access information.
 - [Headings, Subheadings, and Table of Contents \(Google Docs\)](#)
 - [Headings, Subheadings, and Table of Contents \(Microsoft Word\)](#)

DOs and DON'Ts

<p>DO: Provide clear and simple explanations in your own words (be concise)</p> <ul style="list-style-type: none"> • Use short sentences, small paragraphs • Use lists and tables to organize information • Use standard industry terms but explain in simple language • Use AI Tools to help you learn but use lectures first <p>DON'T copy my explanations from the slides as yours.</p>
<p>DO Use Code Snippets and Visuals to reduce lengthy explanations:</p> <ul style="list-style-type: none"> • Include code examples you created • Include comments to describe parts of your code • Include Screenshots, GIFs, Images <p>DON'T use my slides as your image and explanations</p>
<p>DO Include Resources that give steps or help explain concepts</p> <ul style="list-style-type: none"> • You can include resources given in the lecture or find your own but <ul style="list-style-type: none"> ○ Include website link with summary ○ Include links to videos with summary of content <p>DON't use AI to create your technical documentation.</p>

Part 2 Analyze and Apply

Demonstrates a clear analysis of a problem and applies relevant concepts effectively. Work shows independent thinking and understanding. You will use Git, GitHub and Eclipse to edit, version and back up your code.

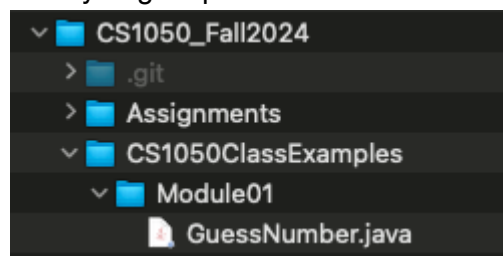
Resources to get started with Git/GitHub and Eclipse

- [Git and Github Version Control](#)
- [Setup Eclipse](#)
- [Eclipse Cheat Sheet](#)

2.1 Explore GuessNumber Code

Be curious about the code as you explore.

1. Complete [Interactive Animation for Guess Number Program](#)
2. Go to you git repo and find the GuessNumber class example in Module01



3. Find the resource to add an existing Java file by dragging and dropping into your CS1050M01 Eclipse project.
4. Add comments to the code to help you understand. You can use parts of the code and comments in your technical document.
5. Version your guess number and technical document locally and then back up on your github cloud server.

1. Take a screenshot showing you are using the debugger to show what is stored in the variable number and guess after you enter a guess.

Name	Value
no method return value	
args	String[0] (id=19)
RANDOM_MULTIPLIER	100
number	11
> keyboardInput	Scanner (id=20)
guess	22

2. Describe 3 things you understand better about the code after using the debugger.

- The debugger does a great job of showing how loop iteration works
- The debugger can help with understanding the timing reassignment of variables
- Helps with showing how a loop terminates after a condition is checked and met / fails

3. Identify the warning in the code.

Show a screenshot of the warning in the program.

```
Scanner keyboardInput = new Scanner(System.in);

// print
System.out.println("Enter your guess:");

int guess;
// Loop
while (guess < 100)
{
    System.out.println("Enter your guess:");
}
```

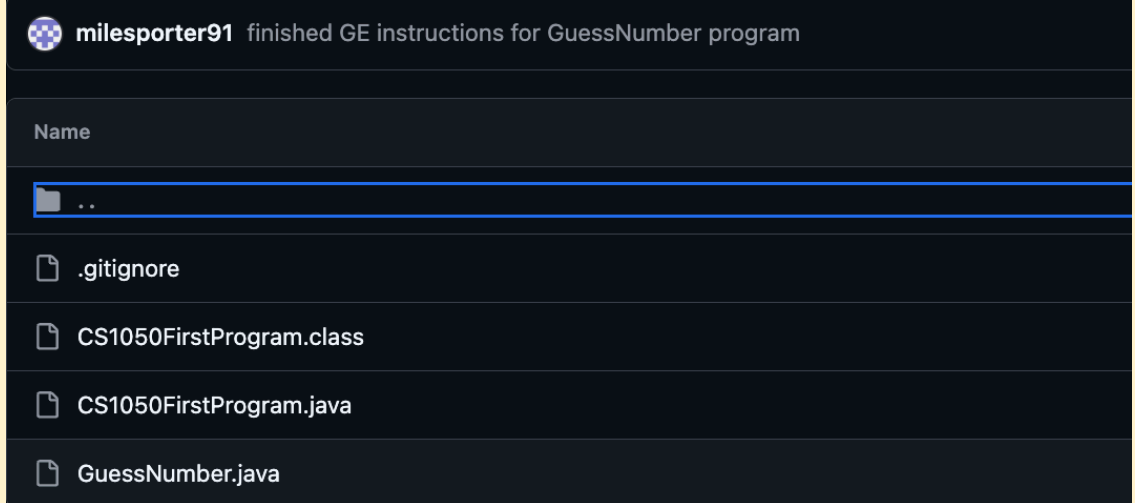
⚠ Resource leak: 'keyboardInput' is never closed

4 quick fixes available:

- ➡ Surround with try-with-resources
- @ Add @SuppressWarnings 'resource' to 'keyboardInput'
- @ Add @SuppressWarnings 'resource' to 'main()'
- ⚙ Configure problem severity

Find a resource to help fix the problem based on the warning message. You can use AI for help with syntax errors and warnings.

4. Take a screenshot to show your GuessNumber.java file in your repo on the Github Cloud Server.



Explain why only the java file is showing in the repos and not the class file.

Because .class is in the .gitignore file, telling Git to ignore files that end in .class or anything else listed in gitignore

2.2 Calculate Grades

You will create a project to store your assignment code and then create your program file. The program you create here will be updated in future guided explorations.

Remember: Commit and push your code often.

1. In your CS1050M01 project create a new class called **LastNameM01Calculations**
2. Apply best practices industry convention for naming variables, constants, methods and classes.
3. Update the code to add an initial comment. **This initial comment is required on all assignments.**
 - a. Your name
 - b. Class name
 - c. Section (M/W) or (T/R) Due date
 - d. Description


```

/*
 * Name: [YOUR NAME]
 * Class: CS1050 (M/W or T/TH)
 * Description: Guided Exploration 01
 * This program will print a couple paragraphs about my passions and interests.
 * The program will calculate a final grade for this class based on the category weights
 */

public class LastNameFirstNameGE01Calculation
{
    {

    }//main

} //Class GE01

```

4. **Explore** different methods to print out information.
5. **Displaying Output:** Add code to the **main** method to print 2 paragraphs about yourself. Some ideas: what you're passionate about, what degree you're working on, what you like to do, what are your goals etc. Only add a little bit of code at a time, run and compile to help you find errors if they occur.
6. **Working with Expressions:** Next add code to calculate your final grade based on the weights in the syllabus. Do not hardcode values in your expression. Use constants for percentage weights and variables for your grades. Remember to use correct naming conventions.
7. Display your name and final grade using printf to format the output to the console.
8. Add comments.
9. Commit your .java code file locally and push your code to github remote repository. Make sure you include comments.
10. Put a screenshot of your code in your github remote repository.



11. Use snippets of code, screenshots and explanations to answer the following in your technical document.

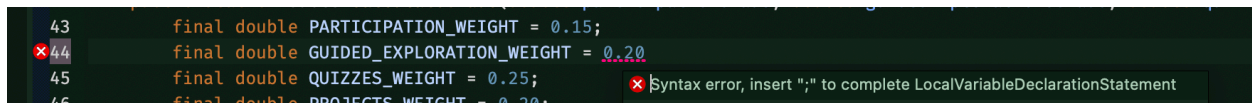
- a. Explain why this would not be a good way to implement your code to calculate a final grade for this class. $\text{double finalGrade} = (.15 * 95) + (.2 * 92) + (.25 * 88) + (.2 * 87) + (.2 * 93)$

This is bad practice because the grades and weights are literals instead of being stored inside of constants or variables that can be references or even adjusted (in the case of variable grades) later

- b. Errors

- i. Explain syntax error. Include a screenshot of your IDE showing there is a syntax error

A syntax error occurs when a user violates the rules of the Java programming language, such as a missing semicolon, curly brace, parenthesis, etc;



```
43      final double PARTICIPATION_WEIGHT = 0.15;
44      final double GUIDED_EXPLORATION_WEIGHT = 0.20
45      final double QUIZZES_WEIGHT = 0.25;
46      final double PROJECTS_WEIGHT = 0.20;
```

Syntax error, insert ";" to complete LocalVariableDeclarationStatement

- ii. Explain logical errors. Explain what you can do to prevent logical errors.

Logic errors are what they sound like; errors in logic in the code. While the code may be syntactically correct, it doesn't do what it was designed to do because of a logic error. For example, performing mathematical operations out of order, or referencing the wrong variables, or less than in lieu of greater than or vice versa.

- iii. Explain runtime error.

A runtime error occurs when the program is running; The code is correct, but some error that wasn't caught by the compiler. Index out of bounds, variables that haven't been initialized being referenced, or dividing by 0 can cause runtime errors.

3 Reflection

1. Identify one specific challenge you faced in this exploration and what you did to overcome it.

Following directions closely, as I did not ignore all the same extensions in git as the instructions said to the first time around, as I went with what I usually ignore on my projects. Because of that, I didnt ignore class files. I fixed this when I came upon a question about it later in the GE.

2. Team Collaboration

What is one strength of yours when working with your group in class?

That I have over 500 hours programming Java in Eclipse, so I can help with application of concepts.

What is one strength of your team?

I don't think we have a team yet, so I don't fully understand how to answer this question.

What is one area you would like to improve about your team working together in class?

See above...same answer

3. Environment Set Up

On a scale of 1 to 5 rate yourself on using the following tools and include one thing you understand and one thing you need help on.

git/github rating: 3

- I understand how to do most basic tasks on GitHub using command line
- I need help on continuing to expand my knowledge of the vast expanse that is Git and Github

If you are not using Eclipse include what IDE you are using

Eclipse rating: 5

- I understand how to do most things in Eclipse, as it is my preferred IDE for over 500 hours of programming in Java
- I need help on nothing that I can think of within Eclipse

4. Now that you are submitting your first Guided Exploration

- What do you wish you would have asked for help on or what did you get help on? I felt pretty confident on most of the material covered, as it is mostly pretty basic programming skills.
- How was your time management? Did you have to do a lot the day it was due? I could have done better about time management, I waited until the due date to even start working on the code. But I only did so because I knew the requirements and that I could get the code done very quickly.
- What will you keep the same and what will you try to improve on when you do the next GE? I will do better about managing my time from here on out, because I am sure the subject matter will become more complex and I will need to give myself the opportunity

