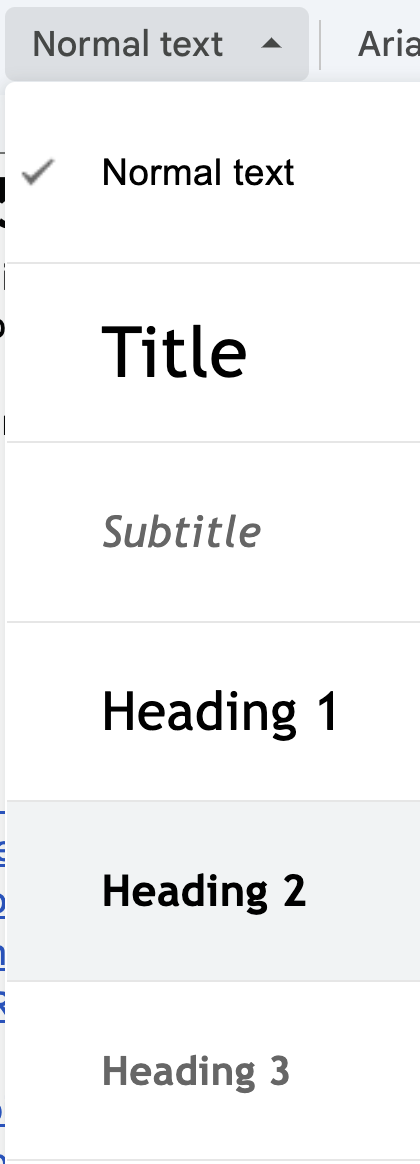
CS1050 Technical Documentation

Use headings to organize topics and create a table of contents to be able to quickly access information.

After adding new headings to organize your documentation, refresh the table of contents below.

[Overview of Developer Technical Documentation](#_e7hnk3gmh67o)

[Set Up Development Environment](#_efvt7vjb7dhk)

[General Resources](#_j8bsu2nohj85)

[Module 01: Fundamental Programming Concepts](#_qn332jwlduxf)

[Compilation and Errors](#_8u1ro2aazjxd)

[Primitive Data Types, Variables, Constants, and Naming Conventions](#_v362y395npbk)

[Arithmetic Operators and Combined Assignment Operators](#_ekcaf49ktnl3)

[Memory Allocation, Conversion and Casting](#_t5qvme324700)

[Code Sample](#_peu7sjhsw2la)

[ANY OTHER TOPIC?](#_9oztm0i5zl6a)

[Module 2](#_qke7paye0rm)

[Module 3](#_345uou64znum)

[Module 4](#_zch7xf7pkk1i)

[Module 5](#_chmkbp3yv5ov)

Follow the 4 tips from [Part 2: Developer Technical Documentation](https://docs.google.com/document/d/1Ve-3OD9EN9DCufeGZTyH_vIt9bsDzUMUv_p8KI1CXlQ/edit#heading=h.4lr6hykc4ng6):

Importance of Being Concise:

* Developers need clear and precise answers.
* Use short sentences, small paragraphs, and simple language.
* Avoid unnecessary complexity to improve clarity.

Avoiding Technical Jargon:

* Simplify writing to ensure accessibility.
* Use standard industry terms if technical language is necessary.

Using Code Snippets:

* Show what the product can do with practical examples.
* Elevates documentation quality by catering to developer familiarity with code.

Role of Visuals:

* [Screenshots](https://docs.google.com/presentation/d/1i8RVexeOWe4fjXEOdkPkMWS1y7S9Mh0ijHz-ds9F5kU/edit#slide=id.p32), GIFs, and videos enhance engagement and understanding.
* Visuals simplify complex processes and reduce lengthy explanations.

# Overview of Developer Technical Documentation

I use headings to help organize and easily access information. I like to include snippets of code that contains comments as an easier way to have documentation. You can put in the document what is helpful for you.

Use information from class lectures, guided exploration, code examples from class and book to be able to use later to help you and approach it as if you could help someone else.

GE 01 Put technical documentation below to demonstrate your understanding of Module 1:

* Understand and set up the programming environment (Eclipse, Git, and GitHub).
* Apply version control using Git and document changes effectively.
* Write basic Java programs using variables, constants, and logical structures.
* Debug code using Eclipse IDE and reflect on common syntax and logical errors.
* Develop concise and structured technical documentation.

# Set Up Development Environment

Provide the following but you can copy and paste my steps and add any additional or update my information for you. Make your documentation your own but follow best practices. Create headings that make sense to you.

* **Overview of tools** 
  + Eclipse IDE for Java development
  + Git and GitHub for version control
* **Git and Github**
  + Overview of tools
  + Steps to set up
  + Steps to use desktop ( or command line if you know it) to version locally and back up on server
* **Set Up IDE (Eclipse):**
  + Overview of you IDE
  + Steps to install
  + Configure the workspace to store projects.
  + Verify JDK installation.
  + Creating future projects and class files
  + Use IDE for debugging -

# General Resources

Here you can list resources that you use frequently and keep adding to them over the course of the class (lectures, book, tutorials, etc)

* [Shared student CS1050 resources](https://drive.google.com/drive/folders/1HvYY8zzSwlsH--03olvqOJooGnJkZ7F4)
* [ADD OTHER RESOURCES]

## 

# Module 01: Fundamental Programming Concepts

Add information for each subheading below. I have some information to get started

## Compilation and Errors

For examples here are some code resources on errors

* [SyntaxErrors](https://liveexample.pearsoncmg.com/liang/intro12e/html/ShowSyntaxErrors.html)
* [RuntimeErrors](https://liveexample.pearsoncmg.com/liang/intro12e/html/ShowRuntimeErrors.html)
* [LogicErrors](https://liveexample.pearsoncmg.com/liang/intro12e/html/ShowLogicErrors.html)

## Primitive Data Types, Variables, Constants, and Naming Conventions

This website describes primitive data types [Java Data Types - javatpoint](https://www.javatpoint.com/java-data-types)

* boolean data type
* byte data type
* char data type
* short data type
* int data type
* long data type
* float data type
* double data type

**Variables and Naming Conventions**

Store a value that may change (vary), have a data type, stored in computer memory,

have scope - defines

● Naming a variable:

Use meaningful names

Begin with lowercase Multiple-word

identifiers camel case

● Declare and Initialize: can declare

and initialize variables in one step

Declare variables by defining what data type (amount of memory needed to store the

value)

Initializing a variable: assign an initial value to variable (before it is used at all).

### 

**Constants**

## Arithmetic Operators and Combined Assignment Operators

## Memory Allocation, Conversion and Casting

## Code Sample

Add code from the guided exploration that contains comments - notes to help you understand the parts of the code.

|  |
| --- |

## ANY OTHER TOPIC?

# Module 2

[ADD Headings 2 for the different topics then add your documentation from the lectures]

# Module 3

[ADD Headings 2 for the different topics then add your documentation from the lectures]

# Module 4

[ADD Headings 2 for the different topics then add your documentation from the lectures]

# Module 5

[ADD Headings 2 for the different topics then add your documentation from the lectures]