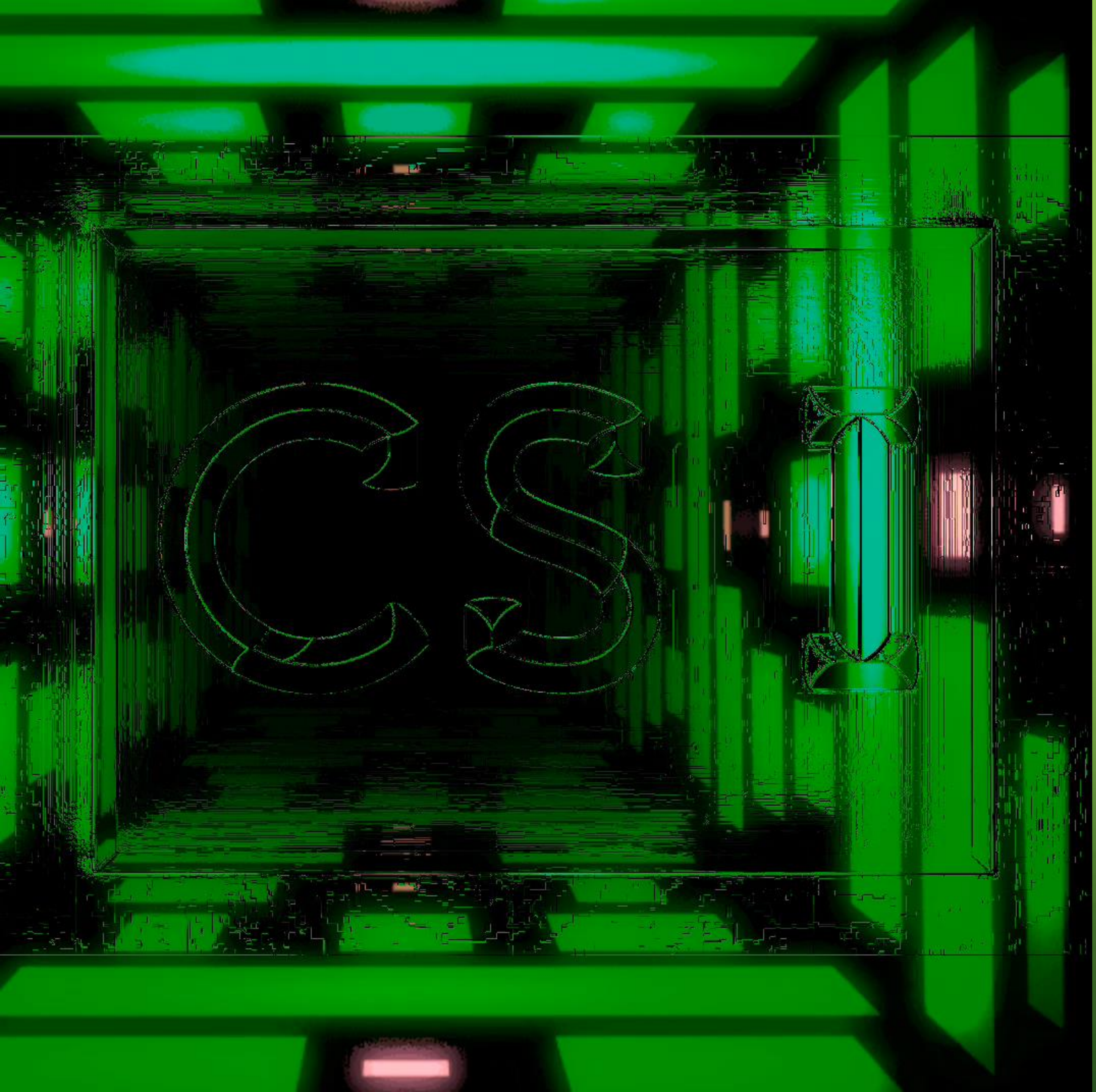

TYPES, VARIABLES, & CONSTANTS IN JAVA

TYPES IN JAVA

JAVA IS WHAT IS KNOWN AS A STATICALLY TYPED LANGUAGE. WE MIGHT ALSO USE THE WORD ‘STRICT’ WHEN REFERRING TO JAVA’S TYPING. TYPES ARE EXPLICITLY DECLARED AND ARE NOT DYNAMIC.

IN SHORT, EVERYTHING IN JAVA HAS A TYPE, AND THAT TYPE IS DECLARED FROM THE VERY BEGINNING AND CANNOT CHANGE.

COMMON DATA TYPES

- int
  - numerical
  - represents an integer
- double
  - numerical
  - represents a real number
  - i.e. a decimal number
- char
  - represents a single character
  - indicated by a character contained in single quotes
  - e.g. ‘a’ , ‘1’, ‘+’, etc…
  - is the sub-component type of String
- Boolean
  - represents a truth value
  - i.e. true or false
  - binary, only two possible values.
- String
  - stores what is known technically as a String Literal.
  - String Literal is contained by double quotes.
  - e.g. “Audra”, “computer”, “7%6=1”, etc…
  - Not a primitive type.
  - A complex, or aggregate type, a String is a collection of chars.
  - note that the keyword begins with a capitol letter.

WHAT IS A VARIABLE?

- a variable is where we store values that change over time in our program.
- a variable has a name that is unique (within its scope)
- special reserved words in the programming language cannot be used as variable names. e.g. can't name a variable int.
- a variable name cannot begin with a number.
- a variable name cannot have any spaces.
- a variable name can contain letters, numbers, and some special characters.

VARIABLE DECLARATION & ASSIGNMENT

- the core syntax of a variable declaration is <type> <name>;
- to initialize a value to the variable at the same time as declaration, we can add an assignment operation. <type> <name> = <value>;
- after a variable has been declared (or assigned), we can assign it a new value using an assignment statement. <name> = <value>;
- '=' is the assignment operator, and in human language, we will refer to its action using the term 'gets.' e.g "the variable x gets 10." is how we would read "x = 10;"

STYLE:
VARIABLE NAMING CONVENTION

When coding, it is important to maintain a consistent and functional style.

For this class, certain style conventions will be required.

For example, we will be using the 'camel-case' when naming our variables.

- variables names must be in camel-case.
  - must begin with a lowercase letter
  - each word in the name (other than the first) begins with a capitol letter.
- variables names should be descriptive
  - related to the function of the variable
  - makes sense to others, not just the code's author
- e.g. lastName, firstName, divisor, penniesLeft. wholeDollars, etc…

VARIABLE DECLARATION & ASSIGNMENT EXAMPLES

- Single declaration
  - int num;
  - double balance;
- Multiple declaration on same line
  - int divisor, numerator, quotient, remainder;
- Single declaration with initialization
  - int num = 0;
  - double balance = 0.0;

- Initializing a variable (already declared)
  - num = 10;
  - balance = 100.0
- Variables should be initialized before they can be used.
- It is good practice to makes sure that all variables are initialized before the main part of the program executes.

CONSTANTS

Sometimes, we what to store values that cannot be altered. The aptly named feature known as a 'constant' allows us to do this.

Constants are particularly useful for parameters that are set independently of the execution of a program.

They are also used for universal constants that do not change, ever, as they are considered known information external to the program.

- indicated by the keyword: final
- core syntax: final <type> <name> = <value>;
- Note that constants are often initialized at the time of declaration.
- It is very important to use constants when it is appropriate.

HOW TO KNOW WHEN TO USE A CONSTANT?

- is the value a universal constant such as the number of inches in a foot?
- is the value a fixed parameter that will always be the same for any execution of the program. e.g. no matter what the input or variables are?
- is the value simulated input that will be used by the program, but should not be modified so as to accurately simulate input which is out of the control fo the program.

STYLE: CONSTANT NAMING CONVENTION

- constants, like variables, should have meaningful and understandable names.
- for consistency and recognition, constant names will be made up of capitol letters with words separated by an underscore _ .
- e.g. PENNIES_IN_JAR, YEARLY_INTEREST_RATE, INCHES_IN_A_FOOT, etc…