Christian J Barden

Steve Jobs said the computer

History of Computing

CS is a Human invention and branch of mathematics

abacus

Machines help with mechanical tasks

computers take input process data to create output

Bits = 4, 1's 8 4 2 1

Byte = 8 bits - - - - / 0 or 1

Software and Programming

Softwar made of pro language

Syntax gramar

Programming Workflow

1) Developing an algorithm in pseudocode
2) translating pro language
3) compile
4) run

Programs

Expresses Solution in Programming Language

Source code must be in the Syntax of the program

Source code → Compiler → Byte code → Interpreter → o

Error triaging

- Compilation Errors — Compiler is an enforcer of rules / Bouncer
- Execution Errors — passed compiler but no output
- logic Errors — wrong output

function

Hello World System.out.println("Hello World");

structure Hierarchy

Class
└ method

Class and Method of printing code

gives us class

Case Sensitive, Pascal/Camel case

Javac Compiler

Classes Pascal case All Caps

java runtime

Variable Name Camel case begins with first letters lowercase, the rest is uppercase

runs the class file

Class name must be the same as file name

Windows Linux WSL

File directory is a tree structure

1) public means file needs to match class name → If just class it doesn't need to match

args can be any word

The main method, used to run most programs

[redacted] [Public static void must be in same order]

Oracle the Java Language Specifications

Hierarchy

System. (class

out. method

println( )

standard output method

3) make sure the code looks clean, i.e. properly and consistently indented. make sure all spaces are either tab or space (not mixed)

other checks (editing)

1.) make sure the variable, method, and class names make sense.

2) make sure the code looks clean, i.e. properly and consistently indented.

Double quotes "Hi" string literals

Comments // anything //

/*
* blah
multi-line Comment
* blah
*/

Make sure the curly brace matches

working copy → staged → commit → logged

git ≠ github

OSS

VCS

create repository ← this is your project

Directory - means folders

A bucket of Something

Move your file to Git

Chapter 1: Computer Programming

The Goal of this class and what it's like to use the scientific method

* The goal of this class/book is to think like a computer scientist
* Computer Science is considered to be a branch of mathematics
* Computer scientists use computer based languages for computations
* Also, they design things to create systems
* They use the scientific method
* A Main important skill in computer science is problem solving.

1.1) What is a Computer?

- A Computer is considered any type of device that stores and processes data.

{Dictionary.com says it's an electronic device that accepts data for mathematic or logical operations; and displays results.}

- Each computer has a different design but the same hardware.

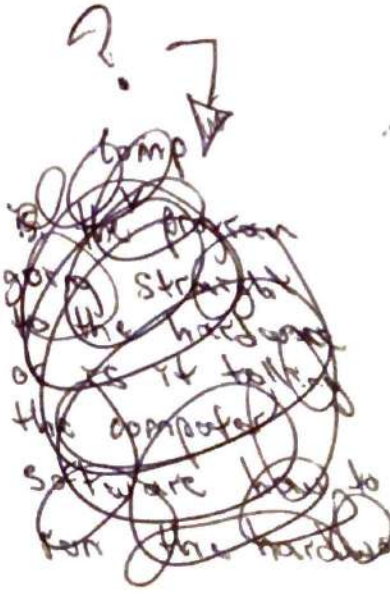
Two of the Most important hardwares

- Processors (CPU) - Performs simple Calculations
- Memory (RAM) - Temporarily stores information

These perform the actual computation

important

64 Gigabytes = 4 billion cells of Memory



1.2) What Is Programming?

A Program is a list of instructions that [redacted] Specifies how to perform a computation on computer hardware.

[redacted]

The Computation can be Mathematical or symbolic like a text.

Ex: Search and replacing text or Compile a program

There are basic instructions that Appear in every Language.

input: get data from file/keyboard

output: Display data on screen/send to other devices

math: perform basic mathematical operations like addition and division

decision: check for certain conditions and execute appropriate code

repetition: perform an action repeatedly, usually some variation

* Think of programming as a large task being performed with smaller subtasks. The process continues until the subtask are simple enough to be performed by hardware circuits.

1.3) The Hello World Program

The first program you learn when programming

- Java programs are made up of class and method definitions.

Statement - A line of code that performs a basic action.

A Print Statement displays a message to its user

System.out.println() displays result on the screen

The name println stands for print line

Confusingly print can display on screen and send to printer.

For understanding purposes, use display when referencing output onto the screen

? Is there a computer language that's not case sensitive

like most times you're ending a statement, the print statement must end in semi-colon(;).

Java is case sensitive, upper and lowercase words have different meanings/actions.

is A Method is a named sequence of statements

Are methods actions that manipulate data?

main is a method
↳ this is special

* Programs can have more than one method.

is a method synonymous with a statement and if that's the case can you call it an action?

1) First you have a class
2) Then a method for running the program
3) Than a method for what the program is going to run

?

A package is a file so it has to be the same as your class right

For now, A Class is a Collection of methods.

The Class has to match the name of the file that it's in. If the Class is Hello, then the file must be Hello.Java.

Java uses curly braces to group things together {and}

The outermost braces contain the class definition, while the inner most braces contain the method definition.

(A line that begins with 2 slashes(//) is a comment which is used for note taking.)

1.4) Compiling Java Programs

- We'll be using a high level language called Java
  - Other High level languages are Python, C, C++, Ruby
- Before you run these High level languages, you must translate it to a lower level called machine language which is a disadvantage

But there are 2 main advantages of a high level language which is

* Easier programming: less writing
* Portable: running on different computers

Two kinds of programs translate high level languages to lower level languages

Interpreters: It reads high level programs and executes whatever it says. (read lines little at a time and perform computations)

Compilers: Reads the entire program and translates it completely before the program runs.

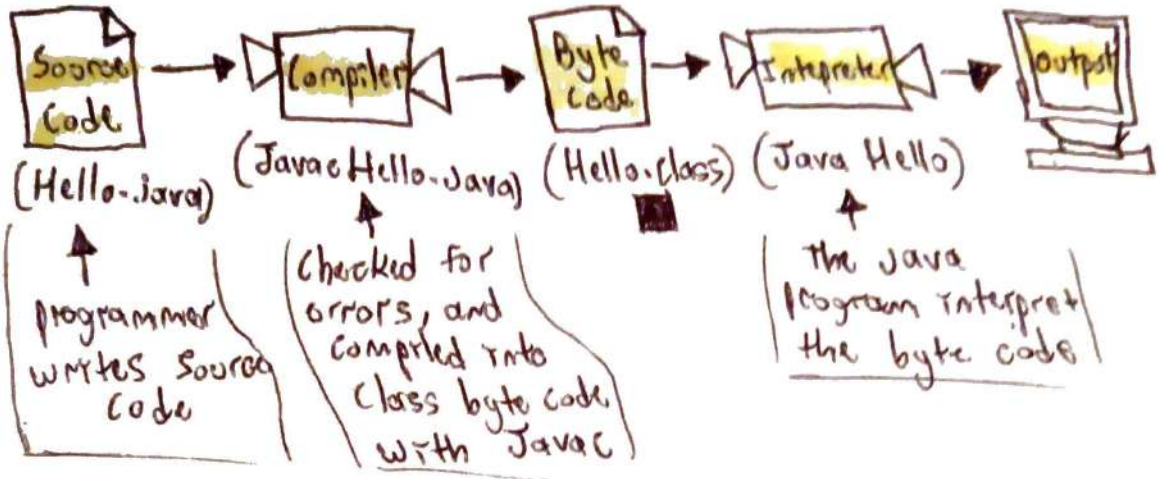
⊙ Compiled programs are often faster than interpreters

- The high level program is called a source code
- The translated program is called an object code or executables

⊙ A compiled executable for a specific machine like windows won't work for all platforms. Therefore, Java is compiled/interpreted code for a virtual machine. The code turns into Java byte code.

low level language. Not portable

- Therefore, it's possible to transfer Java byte code to different machines to run.



[Most IDE do this for you in the background]

1.5) Displaying Two Messages

[Code runs from left to right, top to bottom.]

- Perform two statements of System.out.println() to produce two lines

- Phrases that appear in quotation marks are called strings.

- If you do not want a newline at the end you must use print() not println().

```
Sys.out.print("Goodbye,")
Sys.out.print("Cruel world")
```

output: Goodbye, Cruel world

1.6) Formatting Source Code

In Java source code some spaces are required (at least 1 space between words)

Not legal: public class Goodbye{}

legal: public class Goodbye{}

- New lines are optional

- You can indent your code with tab or (Ctrl+A)

Organizations that do a lot of software development usually have strict guidelines for formatting code.

1.7) Using Escape Sequences

It's possible to display multiple lines of an output with Java you just have to indicate when the line breaks.

Ex: `System.out.println("Hello!\nHow are you doing?\n");`
of new line

This is when you use an escape sequence

Each(\n) is an escape sequence, or two chara of source code that represents a single charac

Common Escape Sequences

\n	newline
\t	tab
\"	double quote
\\	backslash

(Java have a total of 8 Escape sequences)

Ex of double quote: `System.out.println("She like\"you!\"dude.");`

output: she lik "you" dude.

1.8) What is Computer Science?

★ The Main Goal is to Make you think like a Computer Scientist

- Computation is more important than code (Learning)
- You need to decide how to approach problems

An **Algorithm** is a sequence of steps that Specifies how to solve a problem.
↳ Some of these are fast or slow, depending on the memory space it takes up.

Computer Science is the study of Algorithms including discovery and analysis.

Programming errors are called bugs

The process of getting rid of bugs is called debugging

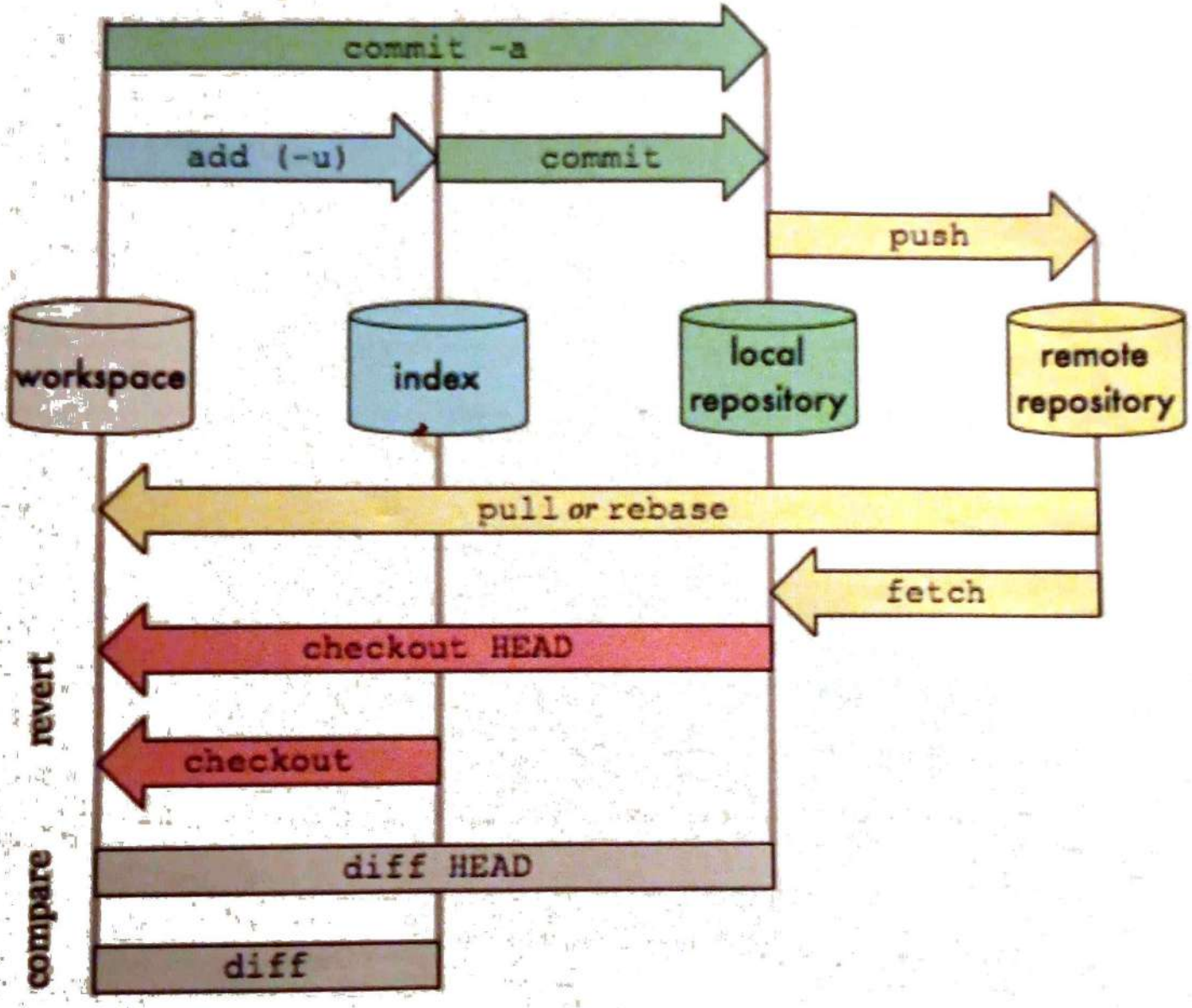
1.9) Debugging Programs

You can try many examples of debugging programs with dr. Java

- Try to make mistakes with new features
- Debugging is an experimental Science if your hypothesis is right you can predict the outcome.
- Programming and debugging should go hand and hand
- Programming can make You Fucking Angry talk to your professor

Git Data Transport Commands

http://osteele.com



Must have a class and main method

pull brings from remote. fetch+check changes

Uploading work through gitbash

1) Make sure you've created SSH Keys

This makes a local copy directory to computer

2) git clone repository url

3) Drag your new files to the newly created repository folder on PC/computer

ls displays what's in your repository

clear cleans console

4) open repository directory with
newly created
cd reponame
on gitbash

5) Then, use git status to verify newly added file for commitment

6) Then, use git add file name to add your file locally to a staging Area
(~~red~~ green)
check git status again it should be green

9) git push pushes your code to github

☺ done

7) git commit -m "changes" of whatever file

(this means you're only commit chang to local com

8) git log will show you all your commit
history

remember the turing machine

Alan Turing discovered Computer Science

1.10) Vocabulary

2. - The output of your source code by your computer is your object

problem solving:
The process of formulating a problem, finding a solution, and expressing the solution

hardware:
The electronic and mechanical components of a computer, such as CPUs, RAM, and hard disks.

processor:
A computer chip that performs simple instructions like basic arithmetic and logic.

memory:
Circuits that store data as long as the computer is turned on. Not to be confused with permanent storage devices like hard disks and flash.

program:
A sequence of instructions that specifies how to perform tasks on a computer. Also known as "software".

programming:
The application of problem solving to creating executable computer programs.

statement:
Part of a program that specifies one step of an algorithm.

print statement:
A statement that causes output to be displayed on the screen.

method:
A named sequence of statements.

class:
For now, a collection of related methods. (You will see later that there is a lot more to it.)

comment:
A part of a program that contains information about the program but has no effect when the program runs.

high-level language:
A programming language that is designed to be easy for humans to read and write.

low-level language:
A programming language that is designed to be easy for a computer to run. Also called "machine language".

portable:
The ability of a program to run on more than one kind of computer.

interpret:
To run a program in a high-level language by translating it one line at a time and immediately executing the corresponding instructions.

compile:
To translate a program in a high-level language into a low-level language, all at once, in preparation for later execution.

source code:
A program in a high-level language, before being compiled.

object code:
The output of the compiler, after translating the program.

executable:
Another name for object code that is ready to run on specific hardware.

virtual machine:
An emulation of a real machine. The JVM enables a computer to run Java programs.

byte code:
A special kind of object code used for Java programs. Byte code is similar to object code, but it is portable like a high-level language.

string:
A sequence of characters; the primary data type for text.

newline:
A special character signifying the end of a line of text. Also known as "line ending", "end of line" (EOL), or "line break".

escape sequence:
A sequence of code that represents a special character when used inside a string.

algorithm:
A procedure or formula for solving a problem, with or without a computer.

computer science:
The scientific and practical approach to computation and its applications.

bug:
An error in a program.

debugging:
The process of finding and removing errors.

SSH ~/.ssh/id_rsa.pub

Secure shell

private key (share)

Move away from master to main

@ Shell scripting

Javac Hello.Word

↑ compiler ↑ source code

- you can create repository locally first

Chapter 2

Workflow

- Find Algorithm
- translate progam language
- Comprle
- hun

```
phublic clas {
  poblic static vod{

}}
```

end every ; statement with

A Method is in a Class

Varrables

- They are labels that reserve space and memory
- The allocation of Values (Very organization Orrented)
- You must declare a varrable before you use them
- Data type describe the world

- Value is on the right

Most Used

- String - int - float
- boolean - double

True/False floating point & fractional

Declaration int i;

Assignment i = 100;

Intialization int x = 42;

Constants final double PI = 3.1459;

Shorthand Assignment Operators

=, +=, -=, /=, %=, *=

Literals

Constant Value in Java

100 an integer

98.6 float

Identifiers

are names of variables and class

Hyper identifiers are camel case (except class)

Chapter 2: Varrables and Operators

This chapter explores how to write statements using varrables which stores, symbols, numbers, words and operators.

2.1) Declaring Varrable

- A varrable is a named location in memory that stores a value.
  - Values can be text, images, sound or other types of data
- A declaration is when a varrable name is declared to a data type.
- Each datatype determines the value that can be stored

```
int x;
```
(means varrable named x can only store whole numbers)

- In general use varrable names that make sense to its function
- The are restricted keywords that can't be a varrable name

^ 50

2.2) Assigning Variables

- Now that we declared our variable we can use them to store values with assignment

```
message = "Hi";
hour = 11
minute = 59
```

- Declarations create storage locations
- Assignment updates storage location values
- A variable must be assigned with what's compatible with its data type.

Ex `message = "123";` legal
(String data type declared)

`message = 123;` Not legal cause it must be a string.

- Variables must be assigned before use
  You can declare and assign variable at the same time or separately

`String message = "Hello";` or `String message;`

`message = "Hello";`

2.3) Memory Diagrams

- Left side of assignment must be a variable name

You can change the value of assigned like this

int a = 5;

a = 21; ← this changes the assigned value from 5 to 21.

2.4) Printing Variables

use System.out.println() to print out your variables

Ex; String FirstLine = "Hello, again!";

System.out.println(FirstLine);

you can also print out characters with your output to make your program more understandable to your users

int a = 97;

Sys println ("you are [illegible] bars old");

2.5) Arithmetic Operators

(Operators are symbols that represent simple computations.

Ex, +, -, * and /

Convert time of day to minutes

```
int hour = 11;
int minute = 59;
System.out.print("Number of minutes since midnight: ");
System.out.println(hour * 60 + minute);
```

(hour * 60 + minute) — called an expression

The replacement of a variable's values is called operands.

2.6) Floating-Point Numbers

Floating-Point Numbers mean to represent values/data with decimals

A typical floating point datatype is double

double accepts decimals/whole number and also outputs a no

- int x = 1.1; this is illegal cause value on right is int and left is double

- double y = 1; legal but bad style make it 1.0

- double y = 1/3; will give you zero since the integer value is zero

- 2.1) Rounding Errors

- Repeating fractions and irrational numbers like pi cannot be stored as a double but must be rounded.

- The difference between the number we want and the floating point / double number we received when rounding is called a rounding error.

instead of
double balance = 123.45;

use this
double balance = 12345;
for accuracy

Floating point arithmetic has benefits that outway the cost but if you need precision use integers.

Integers have more precision than floating point.

2.8) Operators for Strings

- You cannot perform mathematical operations on strings.

- the + operator can join 2 strings which is called contention.

++ = +1

-- = -1

adds together in print

Ex: System.out.println(1+2+"Hello");

output: 3Hello

- Java Executes from top to bottom and from left to right

- When There is more than one operation in a expression things're evaluated/outputted based on order of operation

- If you fuk is still compiling and it say its parsing (being traslated) then some thing was

- Sometimes java tells you exactly what needs fixing.

2.9) Compiler Error Messages

Three kinds of Errors can occur

1- Compile-time errors - Violate rules of java language

2- runtime errors - Crashes that happen when byte code comes out interpretor (usually happens with complex code target)

3- logic errors - logic error is when your program functions but it doesn't do what you (does Not work the way you want)

2.11 Vocabulary

variable: A named storage location for values. All variables have a type, which is declared when the variable is created.

parse: To analyze the structure of a program; what the compiler does first.

run-time error: An error in a program that makes it impossible to run to completion. Also called an "exception".

logic error: An error in a program that makes it do something other than what the programmer intended.

state: The variables in a program and their current values.

memory diagram: A graphical representation of the state of a program at a point in time.

operator: A symbol that represents a computation like addition, multiplication, or string concatenation.

operand: One of the values on which an operator operates. Most operators in Java require two operands.

expression: A combination of variables, operators, and values that represents a single value. Expressions also have types, as determined by their operators and operands.

floating-point: A data type that represents numbers with an integer part and a fractional part. In Java, the default floating-point type is `double`.

rounding error: The difference between the number we want to represent and the nearest floating-point number.

concatenate: To join two values, often strings, end to end.

order of operations: The rules that determine in what order expressions are evaluated. Also known as "operator precedence".

compile-time error: An error in the source code that makes it impossible to compile. Also called a "syntax error".

value: A number, string, or other data that can be stored in a variable. Every value belongs to a type (e.g., `int` or `String`).

type: Mathematically speaking, a set of values. The type of a variable determines which values it can have.

declaration: A statement that creates a new variable and specifies its type.

keyword: A reserved word used by the compiler to analyze programs. You cannot use keywords (like `public`, `class`, and `void`) as variable names.

assignment: A statement that gives a value to a variable.

To assign a variable for the first time.

(Chapter 3) Input and Outputs

- This chapter show you how to utilize your Keyboard inputs to calculate a result, then format that result for output.

3.1) The System Class

Based on System.out.println()

= System is a class that provide methods based on a "System" environment.

- "System.out" has additional methods for displaying outputs

Print out System.out to find out its package origin

The output should be: java.io.PrintStream@

package

- A package is a collection of related classes

(inputs & outputs)

- java.io contain classes of I/O.

- The numbers and letters are the address of System.out which is represented as a hexadecimal.
  - used for rooting and debugging. It addresses

- The address of a value is its location in computer memory.

So "out" is a variable the contains the method println()

- System and PrintStream are classes/files defined in the Java library, which is an extensive amount of class that come with your compiler. Java

3.2) The Scanner Class

The System class provide a special value called System.in

↑ This is an InputStream that has methods for reading inputs from your keyboard. (Not easy to use, but Java has alternatives

An alternative class that Java provides is Scanner which provides methods of inputting words, numbers and other data.

Scanner is provided by java.util, which is a package that has utility classes

- Before you use Scanner you have to import it with `Java.util.Scanner;`
- The import statement tells Java compiler that when you refer to scanner it's defined in the Java.util

In the next line you must declare it before using it.

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

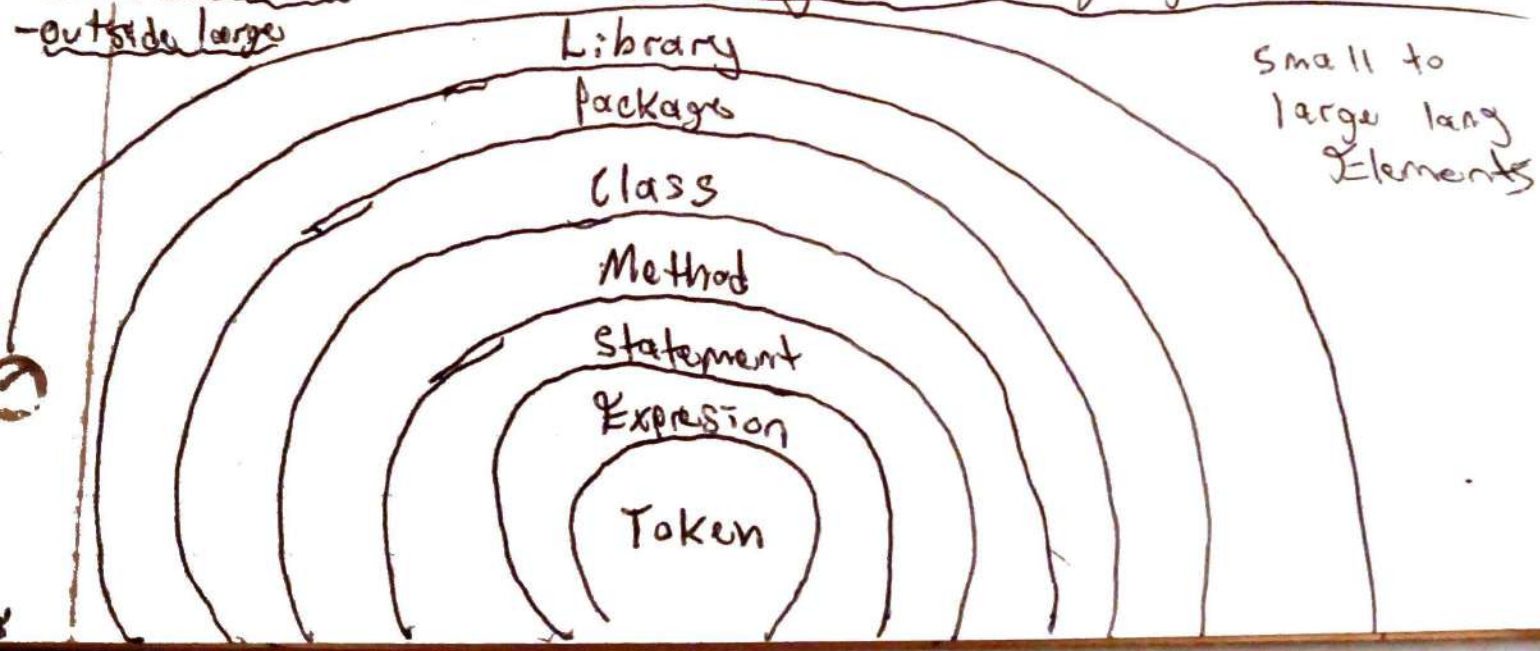
datatype variable method input

Java.lang imports string and other primary classes

3.3) Language Elements

- Inside small/core
- outside large

The Heirarchy of Language Elements



small to large lang Elements

- The Modulos is what you're dividing by.

- Modulo is called the remainder operator

- 76%12 ← this is the Modulos

- if the modulo results with zero (0) they are divisible with one another

Many encryption algorithms use remainder and modulo extracts digits from numbers.

3.9) Putting It All Together

Example

```
import java.util.Scanner;

/**
 * Converts centimeters to feet and inches.
 */
public class Convert {

    public static void main(String[] args) {
        double cm;
        int feet, inches, remainder;
        final double CM_PER_INCH = 2.54;
        final int IN_PER_FOOT = 12;
        Scanner in = new Scanner(System.in);

        // prompt the user and get the value
        System.out.print("Exactly how many cm? ");
        cm = in.nextDouble();

        // convert and output the result
        inches = (int) (cm / CM_PER_INCH);
        feet = inches / IN_PER_FOOT;
        remainder = inches % IN_PER_FOOT;
        System.out.printf("%.2f cm = %d ft, %d in\n",
                          cm, feet, remainder);

    }
}
```

3.10 The Scanner Bug

A String followed by an int works just fine but a int followed by a String has an error unless a new line is in "in.nextLine()" that breaks the code from show characters in a more organized way.

Wrong Way

```
System.out.print("What is your age? ");
age = in.nextInt();
System.out.print("What is your name? ");
name = in.nextLine();
System.out.printf("Hello %s, age %d\n", name, age);
```

Right Way

```
System.out.print("What is your age? ");
age = in.nextInt();
in.nextLine();  // read the newline
System.out.print("What is your name? ");
name = in.nextLine();
System.out.printf("Hello %s, age %d\n", name, age);
```

3.11 Vocabulary

package: A directory of classes that are related to each other.

address: The location of a value in computer memory, often represented as a hexadecimal integer.

library: A collection of packages and classes that are available for use in other programs.

import statement: A statement that allows programs to use classes defined in other packages.

token: The smallest unit of source code, such as an individual word, literal value, or symbol.

literal: A value that appears in source code. For example, `"Hello"` is a string literal, and `74` is an integer literal.

literal: A value that appears in source code. For example, `"Hello"` is a string literal, and `74` is an integer literal.

prompt: A brief message displayed in a print statement that asks the user for input.

magic number: A number that appears without explanation as part of an expression. It should generally be replaced with a constant.

constant: A variable, declared as `final`, whose value cannot be changed.

format string: The string in `System.out.printf` that specifies the format of the output.

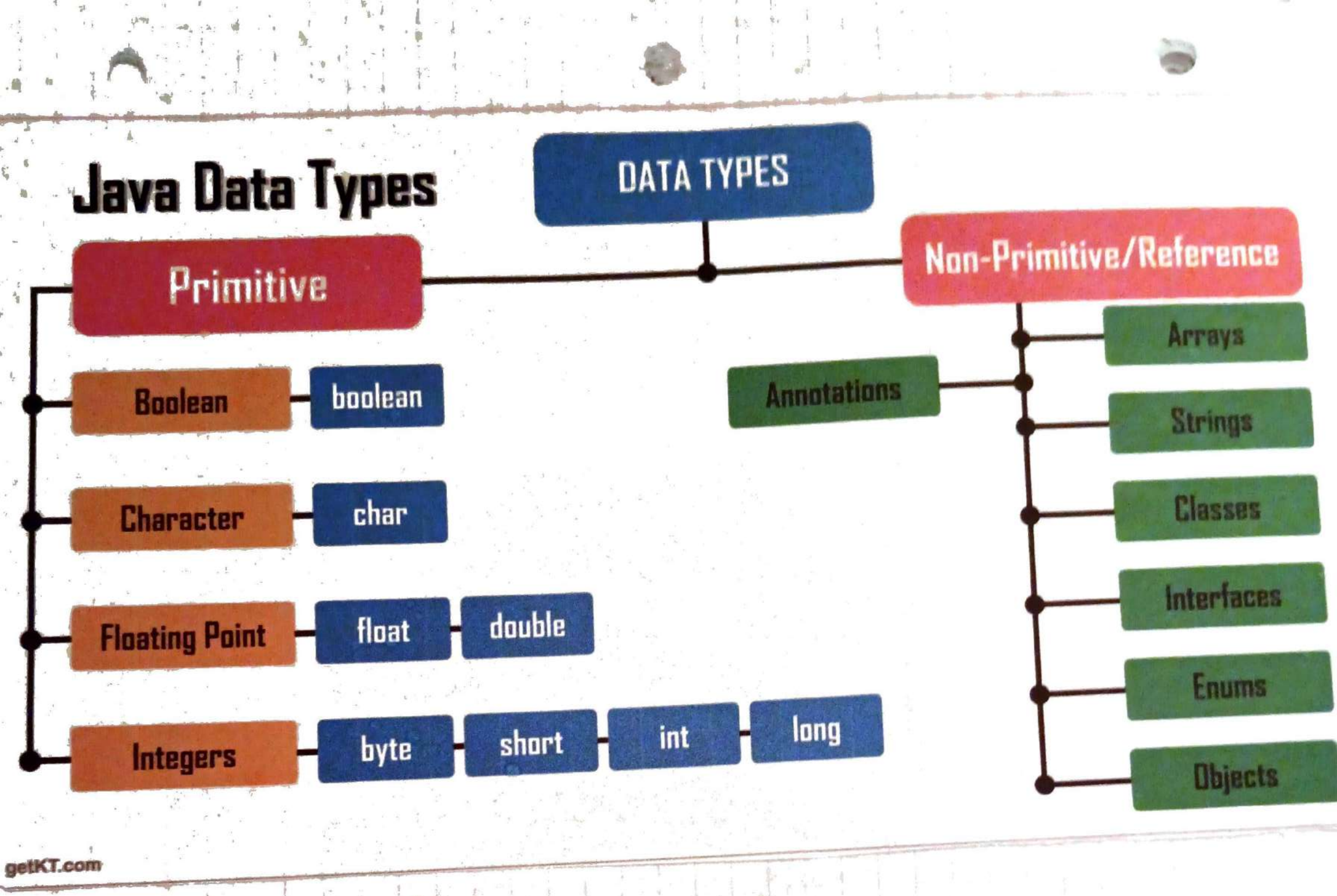
format specifier: A special code that begins with a percent sign and specifies the data type and format of the corresponding value.

stack trace: An error message that shows the methods that were running when an exception occurs.

type cast: An operation that explicitly converts one data type into another. In Java, it appears as a type name in parentheses, like `(int)`.

modulo: An operation that yields the remainder when one integer is divided by another. In Java, it is denoted with a percent sign: `5 % 2` is `1`.

modulus: The value of `b` in the expression `a % b`. It often represents unit conversions, such as 24 hours in a day, 60 minutes in an hour, etc.


Java Data Types
DATA TYPES
Primitive
Non-Primitive/Reference
Boolean
boolean
Character
char
Floating Point
float
double
Integers
byte
short
int
long
Annotations
Arrays
Strings
Classes
Interfaces
Enums
Objects
getKT.com

print stream - output

i/O Stream

byte stream - i/o raw binary data

while

• Character - i/o character data

if

• buffered - optimized input/output reducing api calling

else if
else if

• Scanning - read and format text

else

}

HW

• data - binary i/o

× Every field separated with comma
× Month and day

next line
break
token

input Stream()
A _ 90 _ B18 C

Next()
A 90 B18 C

broken

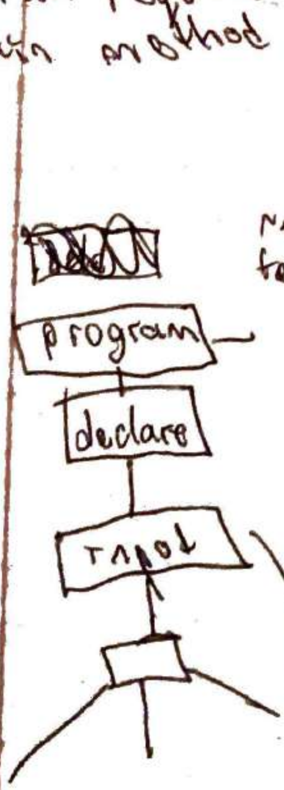
Every Java program requires main method

Object - Binary i/o

Standard streams: system in

System out

when the scanner reads your imported text file how does it show



Scan methods

what's an int or string without declaration

approach programming like a plumber

Input Stream/output Stream

reading

writing

it's easy

Scanner

Print Stream

and 1

Chapter 4:
Methods and Testing

[redacted] Main Chapter points

- How to Organize programs into multiple methods
- Take a look at the Java Math Class
- Discuss strategies for incrementally developing and testing code.

4.1 Defining New Methods

Empty methods perform computations and return results

Ex: nextDouble, which reads keyboard input and returns it as a double

println, return carry out actions without

Java uses the keyword void to define methods

* NewLine produces a newLine like a gap line unless you place parameters to have that line have something in it.

both public static void

* newLine and main are methods that can be invoked from classes and they are both void which means they don't return a result.

* you can contain methods inside void methods to do complex actions

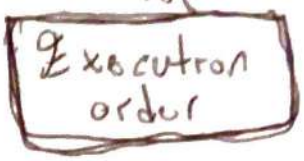
- Methods are camel cased
- Classes are pascal cased

4.2) Flow of Execution

is it a class NewLine program runs methods

- in the opposite order of which they are listed
  - programs begin at first statement of m
  - Statements under invoked methods are invoked first

Executes methods within main from bottom to top




```
public class NewLine {

    public static void newLine() {
        System.out.println();
    }

    public static void threeLine() {
        newLine();
        newLine();
        newLine();
    }

    public static void main(String[] args) {
        System.out.println("First line.");
        threeLine();
        System.out.println("Second line.");
    }
```

Start

end

Statements in threeLine()

↓

statements in newLine invoke println

↓

completes newline

↓

completes threeline

↓

Completes main

No matter what fn it executes at main at the end

⑥ executes from top to bottom in inner methods

Why you should write other methods:

- Creating a new method allows you to name a block of statements, which makes the code easier to understand
- Introducing new methods can make your program shorter by removing repetitive code
- Helps with isolating subproblems

★ Allows you to organize your code separately into individual parts.

4.3) Parameters and Arguments

★ Arguments are the values inside of your parentheses when you invoke your methods.

Ex: println takes a string argument

printf() can take multiple arguments which are formatting string, int, double

has three arguments

Ex: printf("%d in = %f cm\n", inch, cm)

When you invoke a method you provide arguments when you define a method you provide parameters

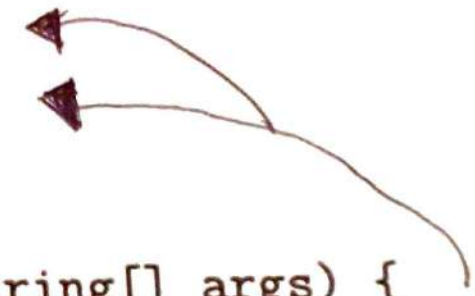
- Parameter passing is when the outside argument is being passed to the inside parameter.

For example:

```
public class PrintTwice {

    public static void printTwice(String s) {
        System.out.println(s);
        System.out.println(s);
    }

    public static void main(String[] args) {
        printTwice("Don't make me say this twice!");
    }
}
```



* Local Variables - Are variables that exist inside specified methods.

Can't use println() for integers only strings

4.4) Multiple parameters

This is an example of a method with two parameters

- You must provide 2 integers to parameter pass into the internal method.

Ex:

```
public class PrintTime {

    public static void printTime(int hour, int minute)
        System.out.print(hour);
        System.out.print(":");
        System.out.println(minute);
    }

    public static void main(String[] args) {
        int hour = 11;
        int minute = 59;
        printTime(hour, minute);
    }
}
```

to (hour1, 0) then passed

(Can be modified and has no effect on main

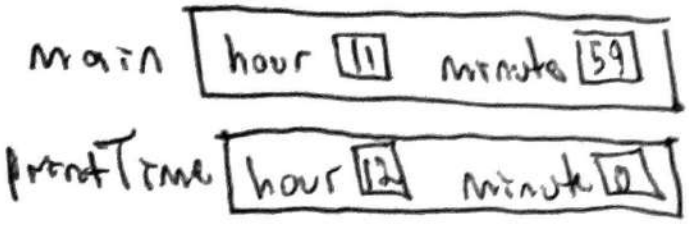
- Even though they have the same name it refers to different variable locations

4.5) Stack Diagrams

(To keep track of variables draw a stack diagram)

A stack diagram is a memor diagram that shows current Methods

- For each method in the diagram there is a box called a frame that contains parameters and local variable of method.
- The name of the method appears outside frame and variables/parameters are inside.



parameter passing
main executes first

Stack diagrams help you visualize the scope of a variable.

Scope is the area of a program where your variable can be used

Learn to trace execution on paper

4.6) Math Methods

- you don't always have to write new methods to get your work done
- The math class provides common mathematical operations.

Common Math methods

Math.sqrt()
Math.sine()
Math.PI
↑ constant

long is bigger than int. In terms of memory

Math.pow()
Math.round
Math.toRadians()
Math.toDegrees()

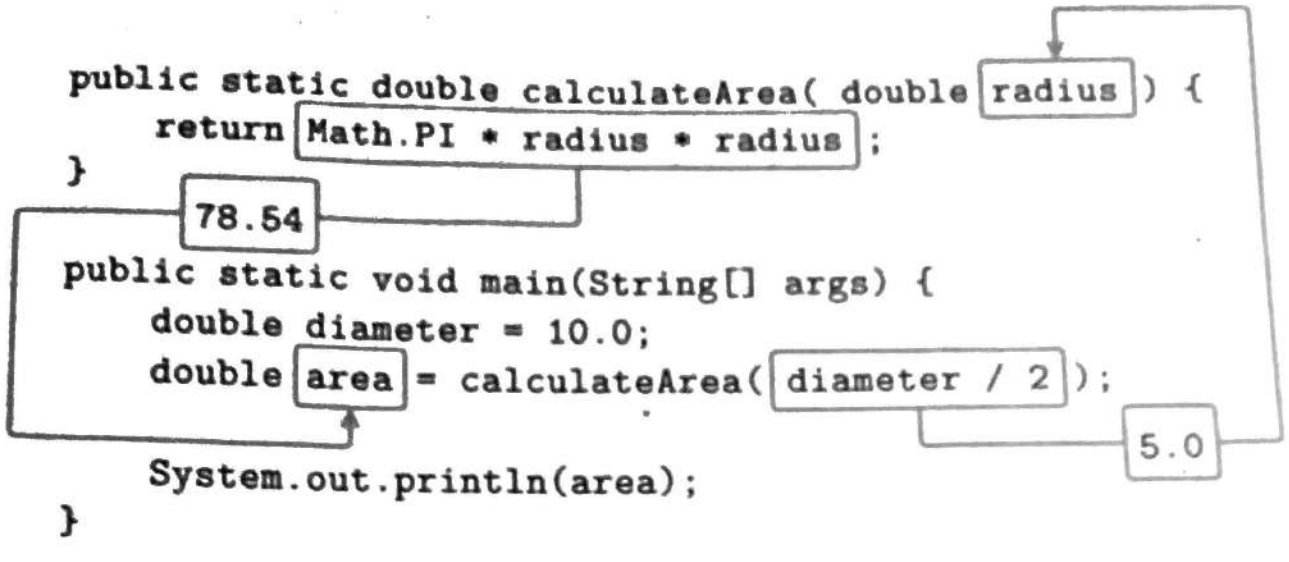
4.0 Composition

- This is primarily detailing that you can put methods inside of other methods a long as the data types match or support by those 2 or more methods

Ex: `Math.exp(Math.log(10.0));`

- exp: base e raised to power of argument
- log: use log base e of 10

remember to include math for calculations people often forget



4.8) Return Values

- When you invoke a void method the invokation happens on a line all by itself
- When you invoke a value-returning method you do an action to a return value

How value return differs from void

- They declare the type of return value
- They use 1 return statement to provide a return value

From top to bottom in main

The return statement must match return type of method.

if return type dou

then method type do

can you declare a datatype to a parameter?

Example of Value returning method

passing the parameter radius and Saving the return value Area

4.9) Incremental Development

People tend to write a bunch of source code ~~without~~ then compile/run it then debug. This wastes time and what they should do is <u>incremental development.</u>

Capital L at the end of long integers

<u>These are the key aspects to incremental development</u>

- Start with a working program and make small changes, if there's an error you will know where to look
- use variables to hold values for you to check
- Keep statements into compound expressions

Scaffolding is when you remove printing statements after done with coding.

<u>Distance Formula in Java</u>

$$D = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

Input = Parameters

Output = return

First Make a method outline called a stub

```
Ex  public static double distance
        (double x1, double y1, double x2, double y2){
    return 0.0; //stub
    }
```

Ex of Distance Formula Code

```
public class DistanceFormula {

    private static double distance(double x1, double y1, double x2, double y2) {
        double dx = x2 - x1;
        double dy = y2 - y1;
        double dsquared = dx * dx + dy * dy;
        double result = Math.sqrt(dsquared);
        return result;}

public static void main(String[] a) {

    double result = distance(0.0,8.0,0.0,1.0);

    System.out.println(result);}}
```

4.10 Vocabulary

void: A special return type indicating the method does not return a value.

invoke: To cause a method to execute. Also known as "calling" a method.

flow of execution: The order in which Java executes methods and statements. It may not necessarily be from top to bottom in the source file.

argument: A value that you provide when you call a method. This value must have the type that the method expects.

parameter: A piece of information that a method requires before it can run. Parameters are variables: they contain values and have types.

parameter passing: The process of assigning an argument value to a parameter variable.

local variable: A variable declared inside a method. Local variables cannot be accessed from outside their method.

stack diagram: A graphical representation of the variables belonging to each method. The method calls are "stacked" from top to bottom, in the flow of execution.

frame: In a stack diagram, a representation of the variables and parameters for a method, along with their current values.

scope: The area of a program where a variable can be used.

composition: The ability to combine simple expressions and statements into compound expressions and statements.

return type: The type of value a method returns.

return value: The value provided as the result of a method invocation.

temporary variable: A short-lived variable, often used for debugging.

incremental development: A process for creating programs by writing a few lines at a time, compiling, and testing.

stub: A placeholder for an incomplete method so that the class will compile.

scaffolding: Code that is used during program development but is not part of the final version.

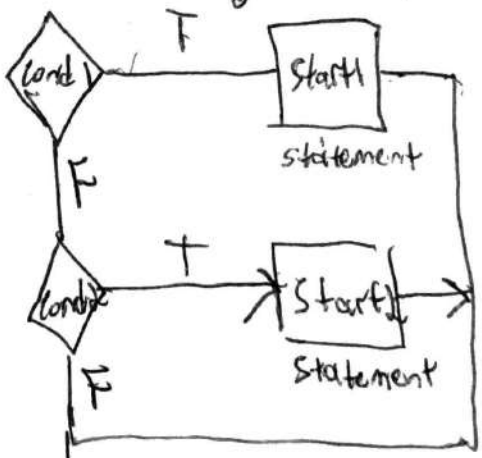
Class Before First Midterm

◇ - decision true/false

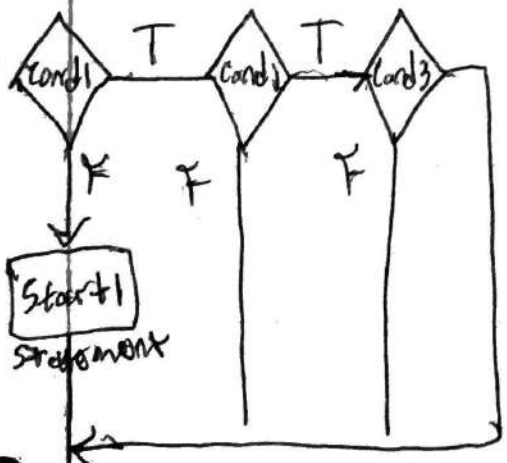
Conditions Flow chart 1

boolean checks

we can branch on true/false



Flow chart 2



* variables are easier to remember large data

conditions on spacecraft

look at this separate from S data

□ = indentation

you can put true/false but dumb — usually has a logical operator

① flow 1

```
□ if (cond1) {
□     Start;   ← println("6")
□ } else if (cond2) {
□     Start2;
}
```

② flow 2 — nested

```
□ if (cond1) {
□   if (cond2) {
□   if (cond3) {
        Start2;
    } printCond3
  } printCond2
} else {
    Start1;
} printCondition1
```

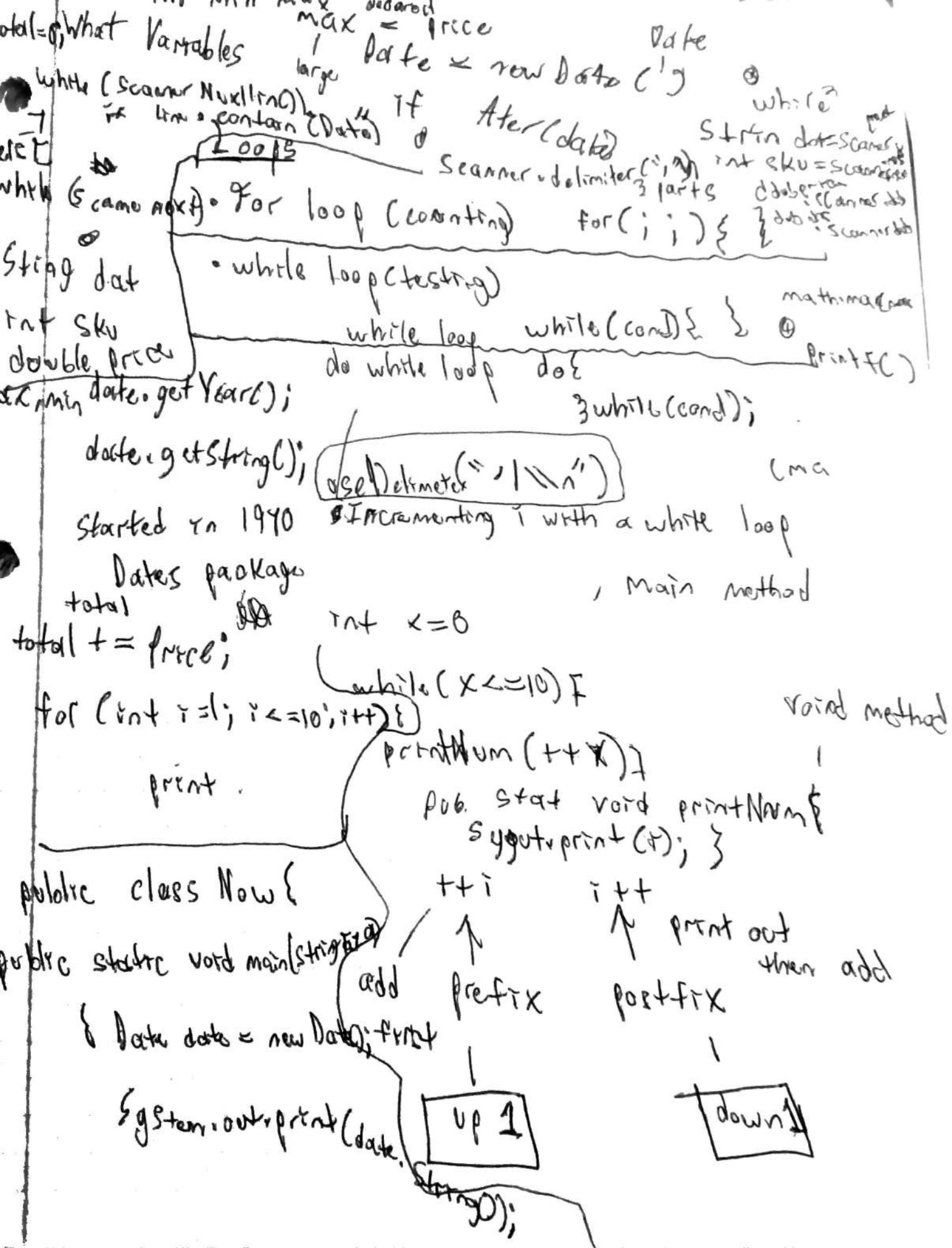
Nested and Not Nested

Truth Table

		A && B (and)	A || B (or)
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

take an artist approach

$$0 \le x \,\&\&\, x \le 5 \quad \text{(or)}$$


Loops
Scanner.delimiter(",");
3 parts
for loop (counting)
for(;;){}
while loop (testing)
while loop
while(cond){}
do while loop
do{
}while(cond);
Incramenting i with a while loop
Dates package
, main method
while(x<=10){
printNum(++x);
public static void printNum{
System.out.print(x);}
++i
i++
print out
then add
add
prefix
postfix
up 1
down 1
public class Now{
public static void main(String[]
{ Date date = new Date();
System.out.print(date.

Chapter 5: Conditionals and Logic

- More Complex programs make decisions on conditions. Therefore, depending on your input a certain output is generated.

5.1) Relational Operators

Java has six relational operators that test the relationship between 2 values

1) X == Y → X is equal to Y
2) X != Y → X not equal to Y
3) X > Y → X is greater than Y
4) X < Y → X is less than Y
5) X >= Y → X is greater or equal to Y
6) X <= Y → X is less than or equal to Y

The result of the relation operator is true or false when using the boolean datatype.

= is the assignment operator it doesn't mean equal

5.2) The if-else Statement

- To write good programs we must write conditions that act accordingly
  - The Conditional statement gives us that ability. The simplest condition statement in Java is an if statement:

```
if (x > 0) { System.out.println("x is positive"); }
```

  - The expression in parenthesis is called the condition.
  - if true the statements in the brace are executed.
  - if the condition is false it moves over to the next block of code
  - A second form is if else statement. The possibilities are called branches, and the condition determines which branch gets executed

```
if (x%2 == 0){
System.out.println("x is even"); } else{
System.out.println("x is odd");}
```

Example

- The braces are optional
However, it's better to use braces.

- No semicolon after condition

- properly indent statements if not using braces, it's called a gotta fart

- Each line of java code should end with semi-colon

5.3) Chaining and Nesting

Sometimes you want to check related conditions and use 1 of several actions. One way of accomplishing this is by a series of if and else blocks:

Outer brach Nesting version

```
if (x>0) {
    System.out.println("x is pos
} else if (x<0) {
    System.out.println("x is n
} else {
    System.out.println("x is zero");
}
```

inner/outer branch

only different

```
else{ if(x<0){
    sysout.ln("
}
else{ sysout(
```

Those are the 2 outer braches nested

- This can be as to as you want but make sure the re standard indented

In addition, you can alternatively use nesting for complex decision making, refer to previous page

5.4) The Switch Statement

- If you need to make a series of decisions chaining else if blocks can get redundant

for example

```
if (number == 1) {
    word = "one";
} else if (number == 2) {
    word = "two";
} else if (number == 3) {
    word = "three";
} else { word = "unknown"
}
```

The Alternative Way is a Switch Statement which is this:

```
Switch (number) {

    case 1:
        word = "one";
        break;
    case 2:
        word = "two";
        break;
    case 3:
        word = "three";
    default:
        word = "unknown";
        break;

}
```

Multiple cases can be grouped

each case ends with a break Ex:

```
switch (food) {

    Case "apple":
    Case "banana":
    Case "cherry":
        System.out.println("fruit!");
    Case "asparagus":
    Case "broccoli":
        System.out.println("veggies!");
        break;
}
```

5.5) Logical Operators

- In addition to relational operators Java has three Logical Operators

&&, || and !
↑ and ↑ or ↑ not

Ex:

- X<0 || X>10, true when either condition is true
- X>0 && X<10, true when both conditions are true
- !(x>0), true when x is not greater than zero

You can use the && logical operator include more conditions

like if(x==0 && y==0){
    print whatever statement or assign this to that
}

Short Circuit evaluation means to ignore the second opperand

5.6) De Morgan's Laws

To Negate a Logical operation you must use de Morgan's Laws

Laws which state to

distribute the not(!) operator to the values in the expression and switch and(&&) or rather or to or(||), and use opposite rational opperators

[redacted]

5.7) Boolean [redacted] Variables

To assign true/flase values you need a boolean variable

flags are use to show if there's an absence or presense of a condition, like if you wanna find out if something is even you check for the output

5.8) Boolean Methods

- boolean Methods can only return boolean values

5.9) Validating Input

- Validating user input is important

You can have an error message displayed if the user place in the wrong input

```
if (!in.hasNextDouble()) {
        String word = in.next();
        System.out.println(word + " is not a number");
    }
```

5.10) Example Program

Chapter 6:

Loops and Strings

ex: searching for document text

Computers are used for repetitive tasks. They do this repetition with almost no error unlike humans.

In this chapter we'll explore how to utilize while and for loops to add repetition to code. We'll also look at string methods and solve some interesting problems

6.1) The while Statement

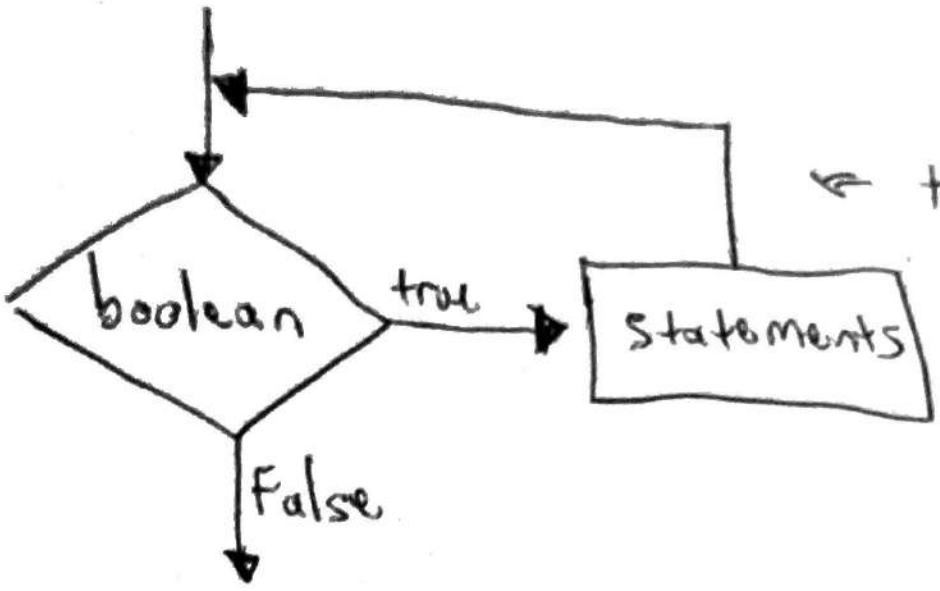
Using a while statement, we can repeat the same code multiple times:

```
int n=3;
while (n>0) {
    System.out.println(n);
    n = n-1;
}
System.out.println("Blast off!");
```

output:
3
2
1
Blast off!

— A space ship reference

(Flow of execution for a while statement)



⇐ this type of flow is called a loop

In the previous example this loop is finite due to the condition not being met a a certain point so it terminates, but if the condition is always met, then your loop won't terminate and it will be infinite.

Checks if program is even or odd and execute statement based on value

6.2) Increment and Decrement

Displays numbers 1 to 5

```
int i = 1;
while ( i<=5){
    System.out.println(i);
      i++;
}
```

shows area to run and terminate

increment $i = i + 1$

decrement $i = i - 1$

6.3) The for Statement

Running the same code multiple times is called an iteration

6.4) We can used nested [redacted] loops to create a multiplication table Ex:

Variables like x and y are called loop variables because they control the execution of a loop

6.5) Characters

- Applying loops to Strings

charAt() stores a character datatype

```
String fruit = "banana";
               012345 — their index
char letter = fruit.charAt(0);
```

output should be "b"

- You can compare characters to rational Operators
  - Character literals appear in single quotes unlike String literals.

C = Alphabet

6.6) Which Loop to use

For Loops and while Loops have the same capabilities and can be rewritten with the same function

- For loops are definite
- While loops are indefinite

it depend on whether your problem is limited of not when using for or while Loops

6.7) String Iteration

- The method.length returns the number of characters in a string

6.8) indexOf method

- tells you the index of a string after a certain appearance like "banana"

int index = fruit.indexOf('a', 2); output: 3

6.9) Substrings

limits string like delimiter but for strings

- fruit.substring(0,3) returns "ban"
- fruit.substring(2,5) returns "nan"
- Fruit.substring(6,6) returns ""

6.10) String Comparison

you cannot use operators on string unless comparing the string length(int) or see if they're the same with boolean

6.11) String Formatting

Format string with String.format(); to return a clean output specified

Quick Heads up

Chapter 5:x(6) Arrays

How to make array

any datatype int[] varname = {x, y, x}

change/Add element position

position
varname[x] = 7;

How to common elements

- var.get(index);

~~varname[];~~

(parse int change data type to int if element is not an integer.)

Christian Barden

(Mon July 25
2- Midterm)

Monday 11th redoc midterm

(Thursday
August 18th
Exam)

- Class Presentation

Greater than 10 is good

New Concepts java.time package.

Contain Time Classes which has time methods.

Deprecated Means no longer supported

H.W. 8 you don't need to use time package

Enumeration is ordered elements

- Classify classes
- use multiple class
- Nested Arrays

No Semi Colon for incrementing

```
for(int N=1; N<=100; N++){
    if(N%3==0){
      System.out.println("Fizz");}
    else if(N%5==0){
      System.out.println("Buzz");}
    else{ System.out.println(N);}

}
```

How to read imported file and Export

access modifier open or close access

```
public static void run(int[][] x){
    for(int row=0; row < x.length; row++){
        for(int col=0; col < x[0].length; col++){
            System.out.println(x[row][col]);
        }}}
```

don't forget method name

add {} // ;

System.out.printf("\n"

for matrix

```
int[] arr = new int[5]
for(int i =
```

// default access modifier is private

```
static int[] createArray(){
    int[] newArr = {1,2,3,4,5};
    return newArr;
}
```

subtract the lowest

for reverse

```
import java.io.BufferReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.FileOutputStream;
import java.io.PrintStream;
public class letsgetBlazed{
 public static void main (String[] args)throws IOException{
        FileReader sc = new FileReader(new File("input.txt"));
        BufferReader xt = new BufferReader(sc);
        String Line;
        FileOutputStream p = new FileOutputStream("output.txt");
        PrintStream k = new PrintStream(p)
    while ((Line = sc.readLine()) != null){
        p.println(); }
}}
```

how to copy File to new file

Christian Barden

1/13/2022.

- There's a function in Java to determine its type
- instance of()
- get type method (local time objects)

How to write your own class?

- Scanner has different constructors
- break program into units of code.
- classes help with organization into even chunks
- Java is object oriented
- Objects and subclasses

Constructor - method of class

Vocab

Class – (blueprint) / cookie cutter – filled with certain properties

Object – object is an instantiation of the instance of a class.

to create an object you would create an instance of a class

→ next page

Christian Barden

1/13/2022

- There's a function in Java to determine its type
- instanceof()
- get type method (local time objects)

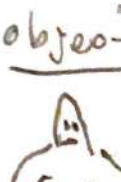
How to write your own class?

- Scanner has different constructors
- break program into units of code.
- classes help with organization into even chunks
- Java is object oriented
- Objects and subclasses

Constructor - method of class

Vocab

- Class – (blueprint) / cookie cutter — filled with certain properties



- Object – object is an instantiation of the instance of a class.

to create an object you would create an instance of a class

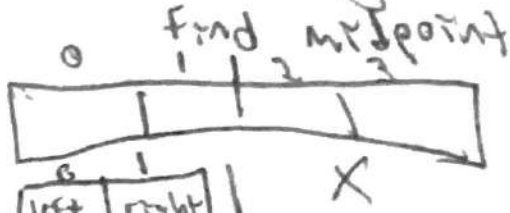
→ next page

Package arrays

boolean found = true;

```
class A{
}
class b{
}
class c {
    class A = new
}
```

Binary search takes a sorted array only



subarray

left

right

(reduce by 2 subarrays)

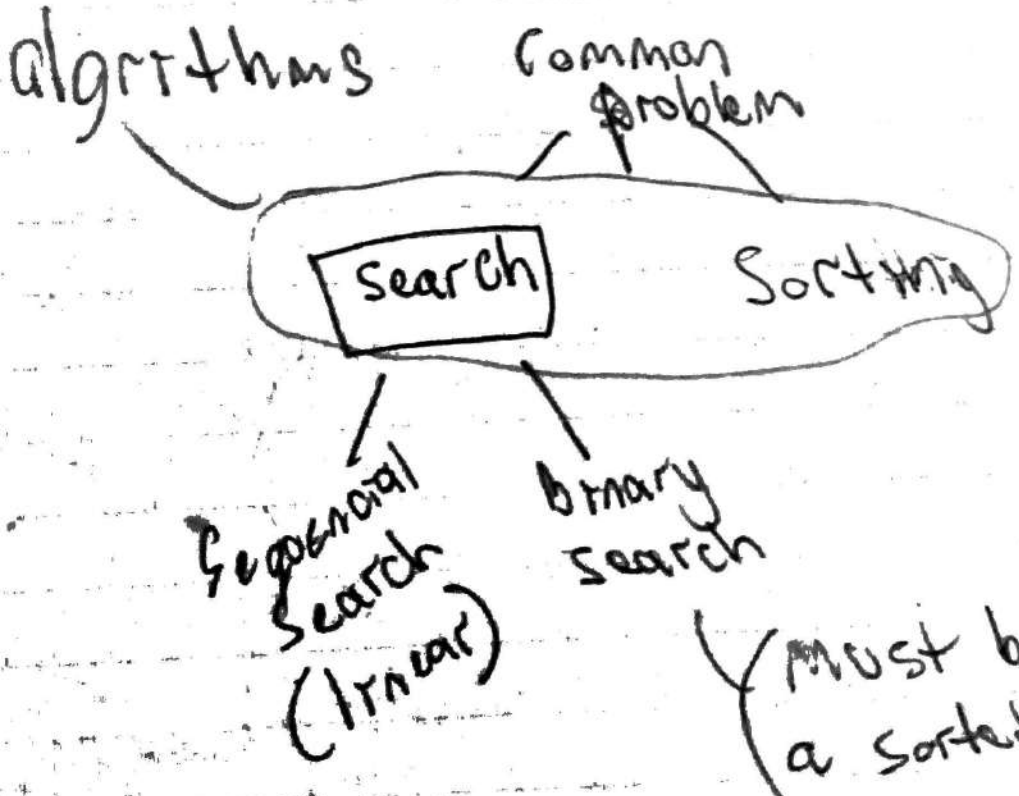
A special Method of a class is constructor

$2^3 =$

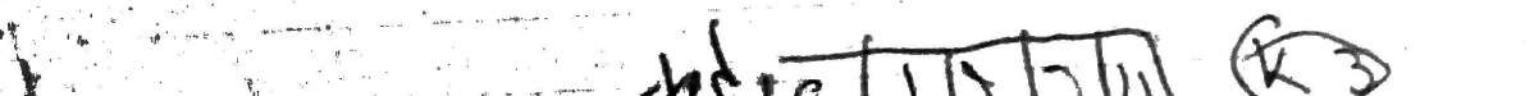
$\log_2(8$

all al

effici



(must be a sorted array)



attempting with recursion

```
public static void BinarySearch(int[] array, int key){

// (0-9), (5-9) //

// System.arraycopy //
    int mid = array.length/2;
    int [] left = new int[mid];
    int [] right = new int[mid];
System.arraycopy(array, 0, left, 0, mid);
System.arraycopy(array, mid, right, 0, mid);

display(left);
display(right);
}

pub        display()
for (int i =0; i< a.length; i++
```

- not initializing values defaults to zero
- assigning

~~int leftlast = left[left.key-1]~~

~~int rightFirst = right[0];~~

~~(rightFirst<=key)~~

~~(k<=leftLast)~~

can store dimensions in variables

```
int[][] matrix = new int[rows][cols];

for(int row=0; row<matrix.length; row++)
  for(int col=0; col<cols; col++)
```

2 Dimensional arrays

sos

methods

```
int token = sc.nextInt
matrix[row][col] = token;
```

```
largest = 0
while(i<arr.length)
if(arr[i+1]>arr[i])
  arr[i]>largest
  largest = arr[i+1]
```

for loop

for loop

place

```
if(arr[idx]>largest)
```

static method when returning method

```
while(idx<arr.length){
  largest
```

7:

```
public int readData(String[] x, Double[] y){
        Scanner sc = new Scanner( new File( ...
        int idx = 0;
          count = 0;
      while (sc.hasNext){      String[] ab = new String...
          ab[0] = sc.nextline
          ab[1] = sc.nextline
          ab[2] = sc.nextline
          ab[3] = sc.nextline

                                  Double [] ac = new Doub...
Double x = parse.double(sc.nextInt);
Double y = parse.double(sc.nextInt);
Double z = parse.double(sc.nextInt);
Double p = parse.double(sc.nextInt);

Double [] ac = (x + y + z + p) / 4
    return count++++;
}
```

Bubble Sort

Sorting algorithm

Operands vs operators

1. Compare two items
2. Swap 2 items

Bubble Sort with Hungarian

Hungarian bubble sort

establish all the largest numbers