JAVA IS VERY STRICT

**Introduction**

The name of the class must be the name of the file – if the class is hello world, the file must be hello world – the java file must have the same name as the class - java might have inner classes with different names, but the main file itself must have the name of the main class.

Java only runs from the main method. Nothing else – once the cold is not executed through the main method, it will not work.

Java is a compiled language and not an interpreted language

When creating the file, always use **a capital letter** to Start

Whatever you do, if you want your code to run, it must be the main method

public class(this is the class) Main (name of the class ) {

public static void main (String[] args] [

System.out.printIn (“Hello world”);

}

}

Java only sees main before it runs – any class at all, it will look for a main method as specified above. it only sees main before it can run . Java only sees the main before it can un

2 types of classes in Java : static and dynamic

Static can be called on the instance of the class

***How to run a Java file***

Navigate to where the file has been saved. Type “Java + name of the file “ just like python

Java has 2 files , its not an interpreted language, it must be compiled before running. So to compile before running, you need to type “javac + name of the file” then type “java + file name ) However, in Vs code, it normally compiles and runs for you- but if you’re gonna be using a normal command prompt then you will need to manually compile and run like shown in this paragraph

You don’t need to have a main in every File, you can import to another file that has a main in it.:

public class Test {

public static void displayresults (String[] args] [

System.out.println (“You are doing well ”);

}

}

public class Main {

public static void main (String[] args] [

Test.displayresults

System.out.printIn (“Hello world”);

}

}

The solution will be:

You are doing well

Hello world

If you don’t wanna run a code, just add “//” in front of the code – it is a way to write a comment or - /\* and end it with \*/

Variables – containers for storing data values –

String (in quotation marks), int (integers) , float, Boolean(true or false – remember in Python) , Char (a single letter is a char)

**2 types of Data types in Value**

Primitive – this just stores values – int, float and Boolean

Non-primitive- Strings, can store methods and even now locations

How do you declare a variable

Type , name of variable, assignment operator and the value of the variable

int examScore = 90;

float floatNumber = 0.5f;

boolean hasPass = true;

If you declare a variable in a method, it is only assessable in a method- it is only going to know the variables in the method that you’ve stated the variables.

**Non primitive**

Test test = null;

Type Java casting:

Basically converting one value to another when using it . That approach from changing one variable to another is what we call type casting

**2 types of Java casting**

Widening casting – if you’re converting a smaller data type to a larger data type

Narrowing – converting a larger data type to a small data type

A Double – a value that can hold more than the basic value – so 24.4434 is a double – so if you’re converting from 24 to 24.667 is widening cast

If you’re converting from 24.666 to 24 is narrowing

**Widening example:**

Int examScore = 34;

double doubleExamScore = examScore’

Print. ln only takes strings so you normally need to convert integers - system. out.println ( String.valueOf(doubleExamScore)

You can call a static method in a dynamic method or another static method , but you cannot call a dynamic in a static method – if you remove “static” from a method, it will not work

**Java Operators**

**Arithmetic operators:** +(add) -(subtract) \* (multiply) / (divide)%(modulus), ++ (increment) - - (decrement)

**Assignment operators**: this is reassigning a variable. If you don’t want to reassign a variable, add final to it

So = int valueB = 2

Int valueB = 4

In the code, valueB will be reassigned to 4

But if you do not wanna change to another thing, you must add “Final” to it.

S0 = **final int valueB = 44** – this means that valueB cannot be reassigned.

Comparison – when you compare, the results are always true or false- it always **brings out a Boolean**

== equal to

! = not equal

> Greater than

< less than

**Logical Operators**

**Conditional Statements**

2 main ways; if or switch

If statements and Switch Statements**.**

If statement runs exactly like an **if** and **else** statement . – there is also **else If** – there is only one if and else but the else if can be a 1000.

The **else if** must come before the **else**

You can use &&

So if examScore >= 70 && examScore > 10)

Both conditions will have to be true before a result will come true

examScore = 70

String studentInitial = “A”

Also, if (examScore >= 70 && studentInitial == “A”)

**It will work**

You can also use **or “||” so if it sees this or that, run this code**

**Switch –** runs like an else statement but it must be highly specific . It matches specific cases – like if the month is = January, do this

You can have a lot of cases

Interations and Loops

Programs are powerful because they can repeat things so fast

There is a for Loop and a while loop

If you know the number of steps to go to, use a for loop but if you don’t know the number then it is a while loop