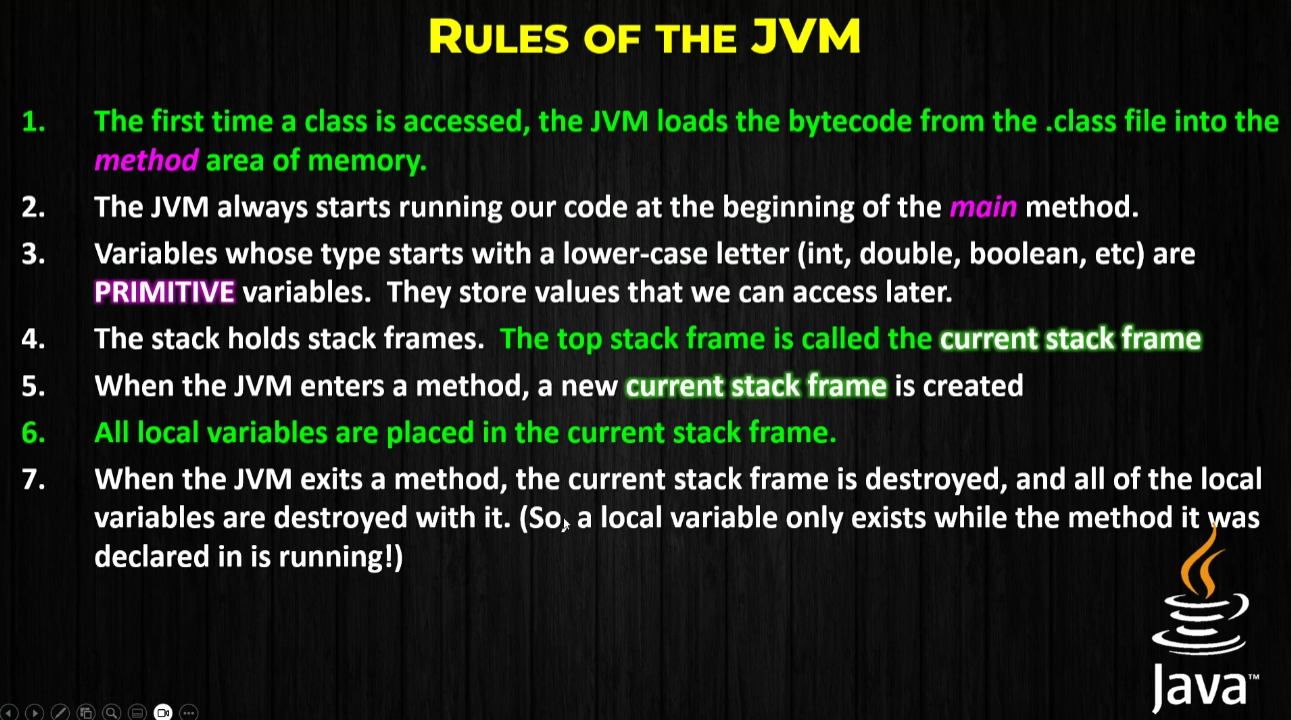
**Java Notes Week 2**

* Local Variable
  + Variable declared inside method
* Stack memory
  + An organized area in memory where stack frames are created and destroyed
  + Every time a method is entered, a new current stack frame is created
  + Every time a method is exited, the current stack frame is destroyed
  + All local variables are stored in stack frame
  + The JVM can always see the local variables in the current stack frame
  + All variables declared inside of that method are local variables and are stored in the current stack frame
* Rules of JVM
  + The first time a class is accessed



How does JVM Run program

* WHen the program starts, the class loader loads the hellowworld.class bytecode into the code section and the main method starts to run
* New current stack frame created every time a method is entered
* Local variables are stored in current stack frame
* Current stack frame (as well as all of the local variables stored in it) are destroyed when the method ends
* Debbuger
  + Must add breakpoint’
  + Allows us to pause our program at any line while running
  + When debugging, program freezes at breakpoint
  + Call stack
* Decision Statements
  + Boolean can be true or false
  + < less than
  + > greater than
  + <=
  + >=
  + ==
  + !=
  + ! not
  + Logical operators
    - && means and
    - || means OR
    - ! means not
    - ^ means Exclusive OR
  + Order of logical operations (BNAO)
    - Brackets
    - Not
    - And
    - OR
  + If statements
  + if (a == 7) { System.out.println(“This code print is boolean expression is true}
  + Vars only exist in the code block they are declared in
* IF statement shortcut
  + output = (num1+num2>10)?"Zoinks":"Narf";
  + If the statement inside brackets is true, Output = Zoinks, if it is false, output = Narf
* Sequence – code runs in order
* Selection: Decision
* Iteration: looping
  + Allows you to not have to repeat code
  + Allows us to run a chunk of code as many times as we want
  + Like an if statement stuck in a loop
  + For while loops, as long as the Boolean condition is true, it will keep running. Top check
* Do while loop
  + Checks condition at the bottom, after executing the loop at least once
* Sentinel conditions
  + A constant value that with the purpose of breaking the loop
* Continue
  + Makes the loop go back to top where the condition is without running any code after continue
* Break
  + Ends loop
* Return
  + Exits main method