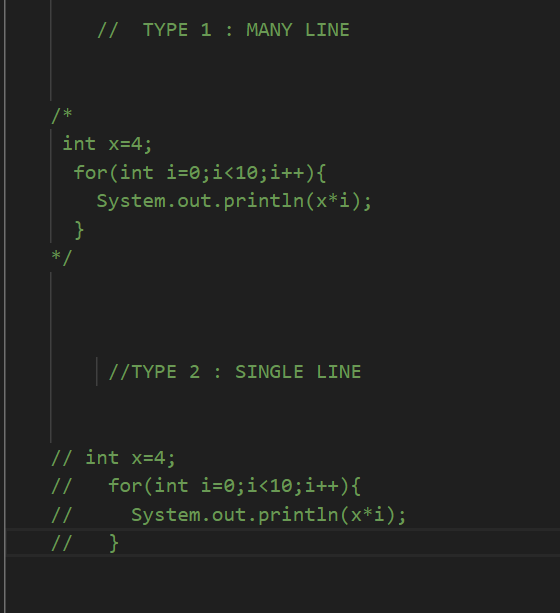
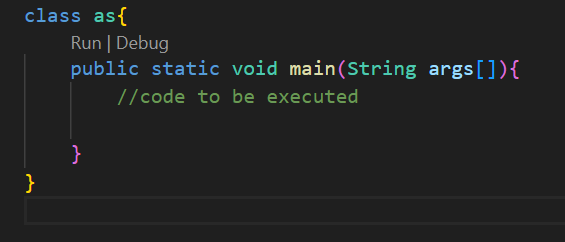
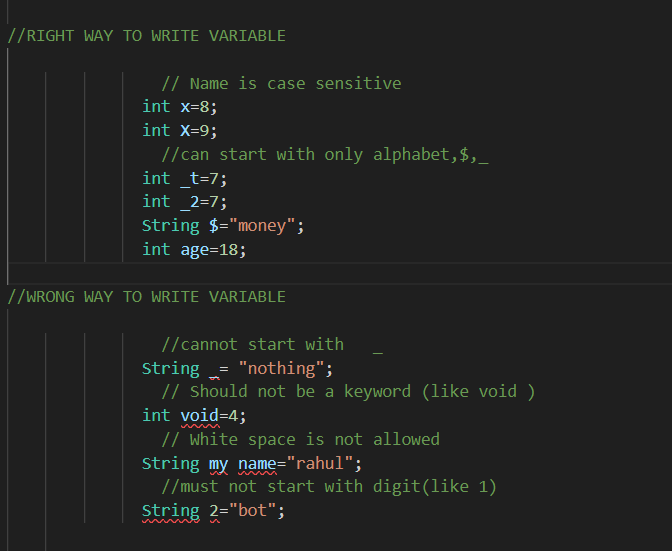
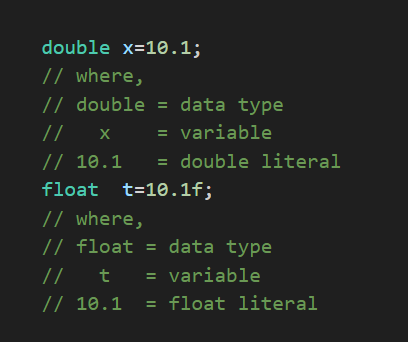
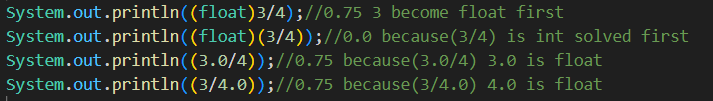
Format Comments

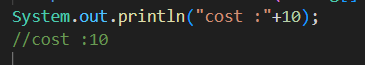
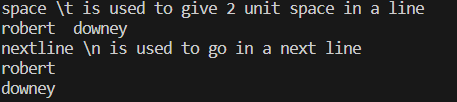


Data Types and variable



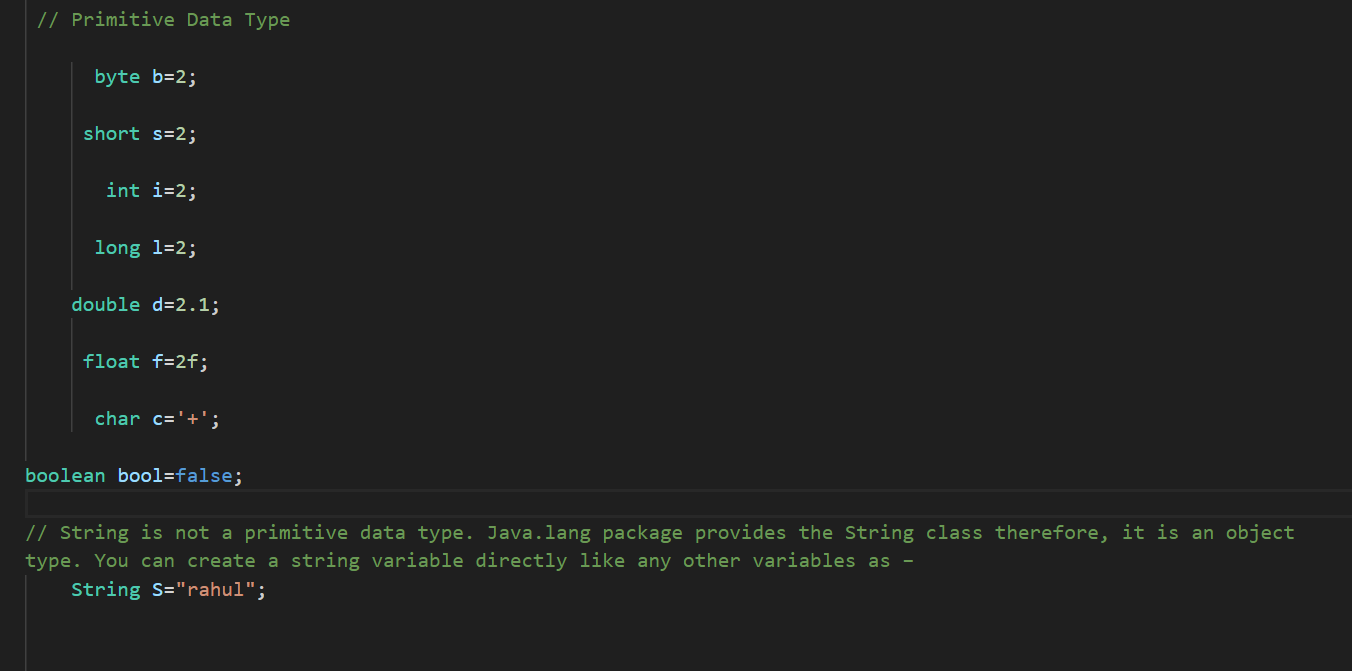
 initializing multiple similar datatype variable



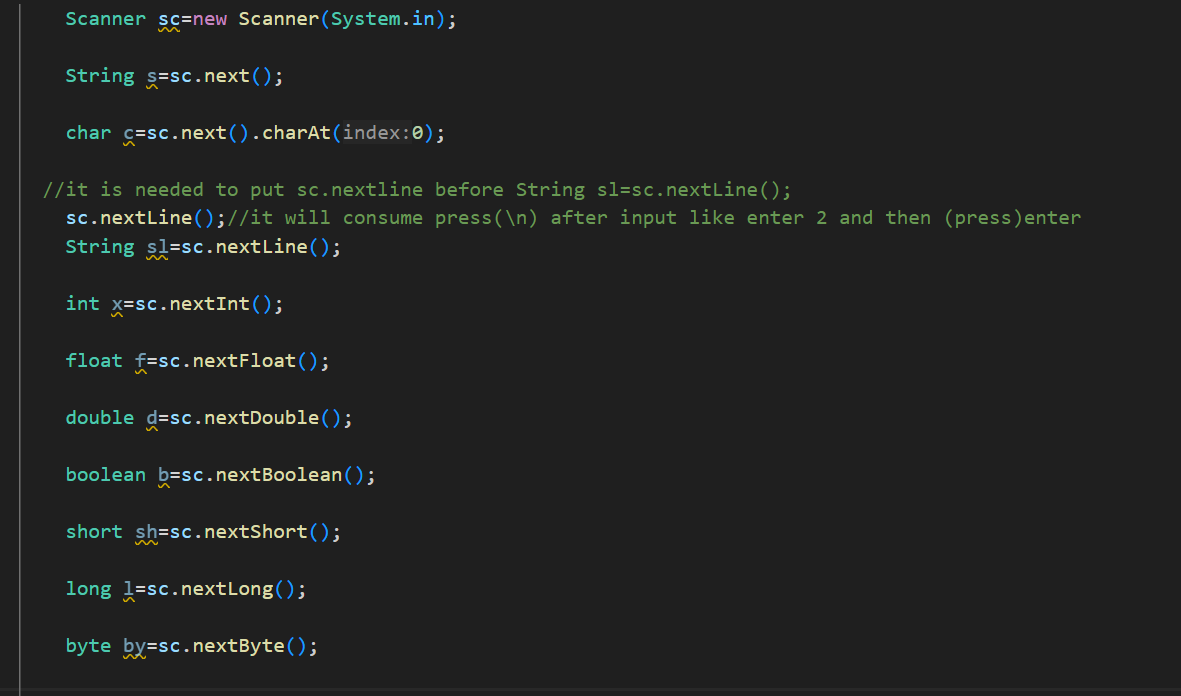
Wrong format to initialize 

Primitive Data Type



Scanner Class

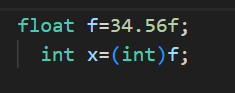
( import java.util.Scanner)



Data Type Conversion /widening/implicit conversion

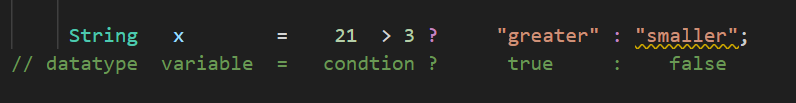
Byte🡪short🡪int🡪 long🡪 float🡪double

Type Casting / Narrowing /Explicit Conversion



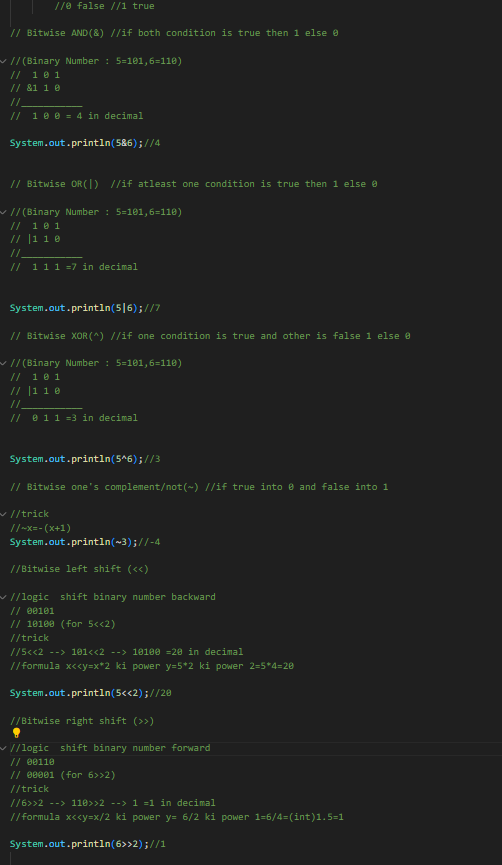
Operators

1. Arithmetic
2. Binary (mathematical) operator: + - \* / %
3. Unary operator: ++x x++ --x x—
4. Ternary operator: ?:



1. Relational ( == != > >= < <= )
2. Logical ( && || ! )
3. Assignment ( = += -= \*= %= /=)
4. Bitwise (
5. &(bitwise and)
6. |(bitwise or)
7. <<(shift left)
8. >>(shift right)
9. ~(one’s complement)
10. ^(bitwise exclusive or) )

Bitwise Operator in detail



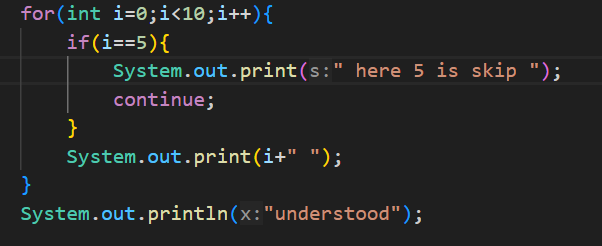
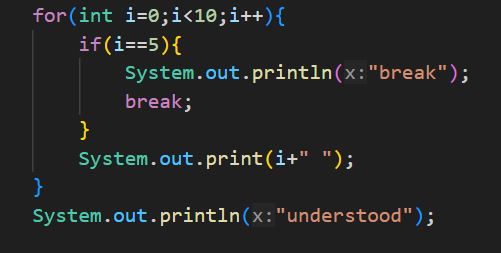






Break and Continue Statement

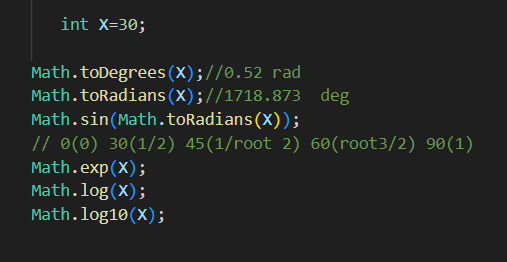
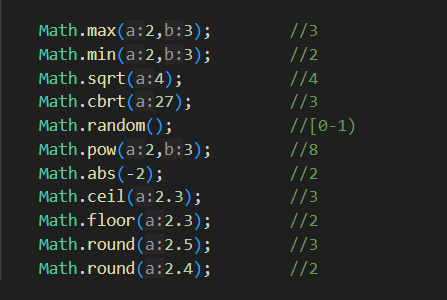
Break(to exit loop) Continue(to skip specific condition iteration)



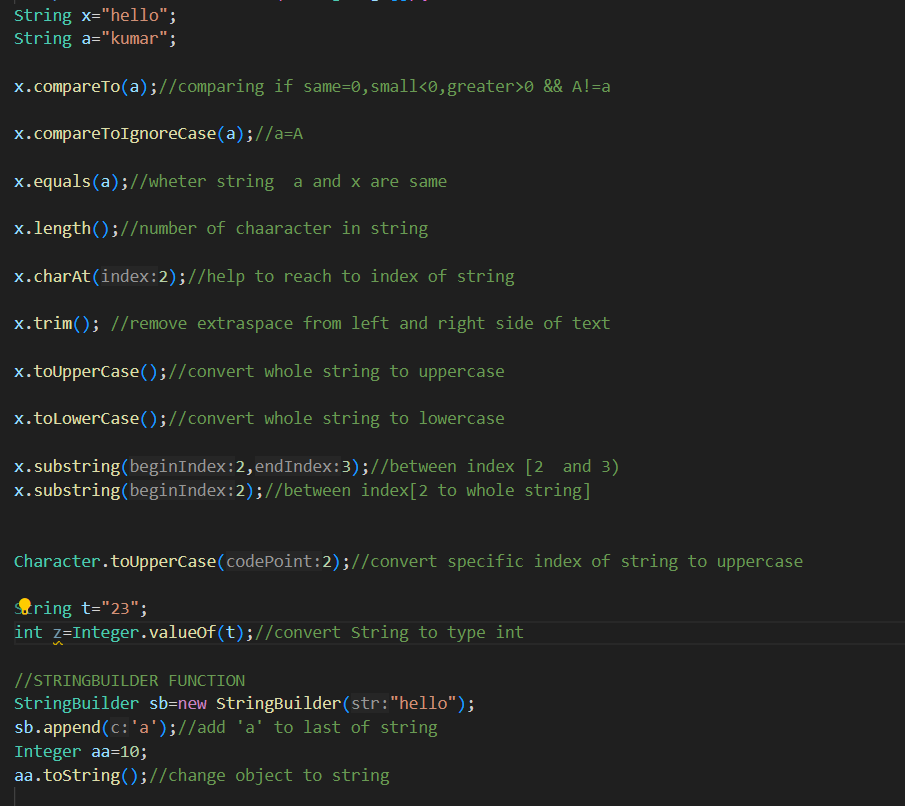
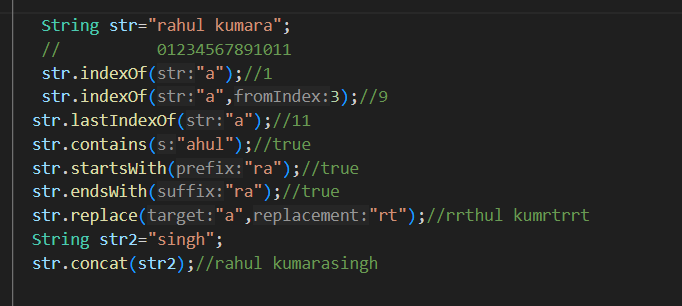
***Output: break Output: continue***

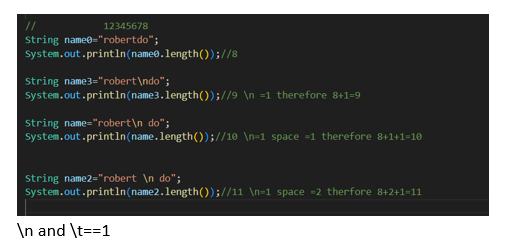
******

Math function



String Function



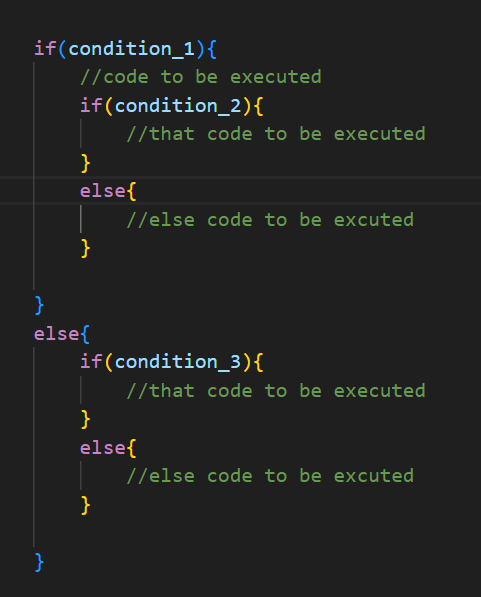
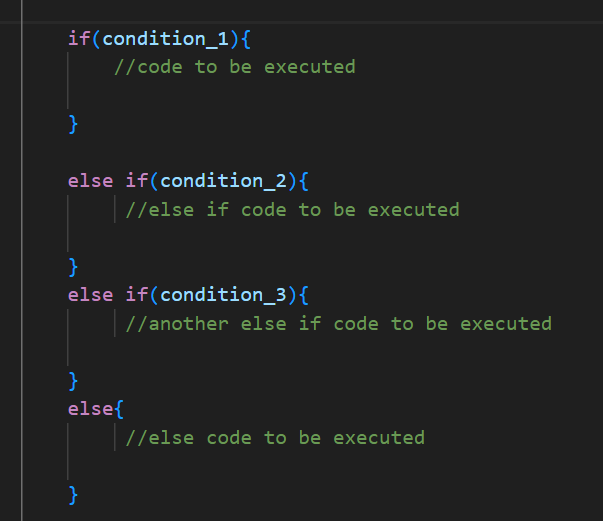


Conditional Branching/Selectional Control/Decision Making

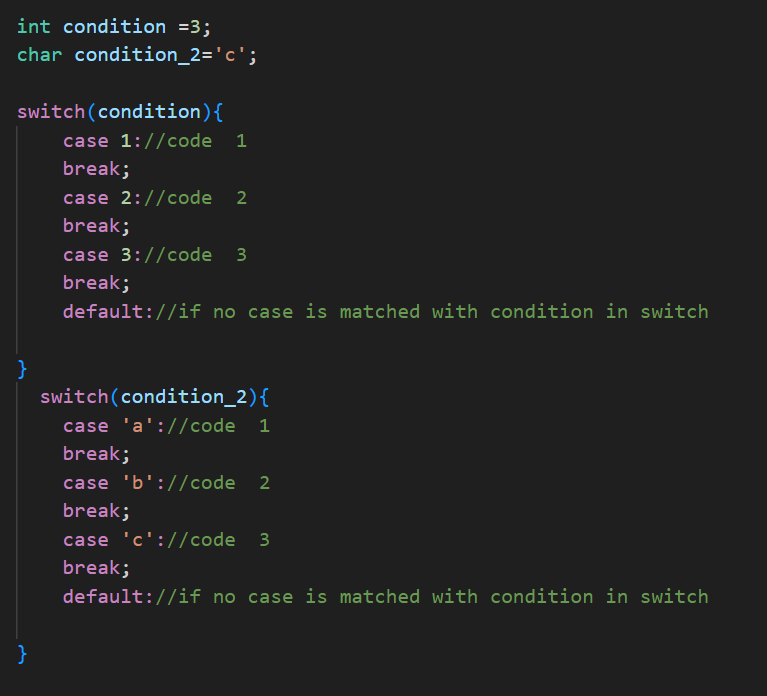
# If Statement if else Statement

# 

# Else if Statement Nested if Statement

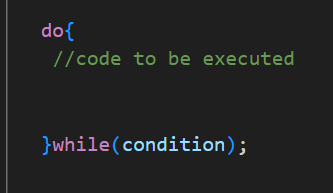


# Switch statement

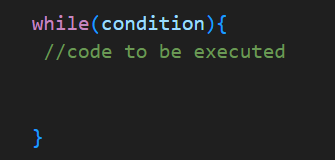
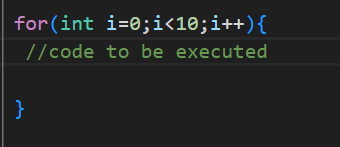


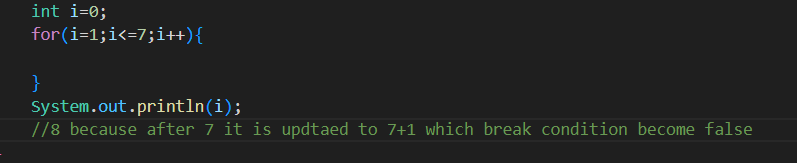
Loop Statement

# 1)Exit Controlled Loop/post tested loop (do while loop)

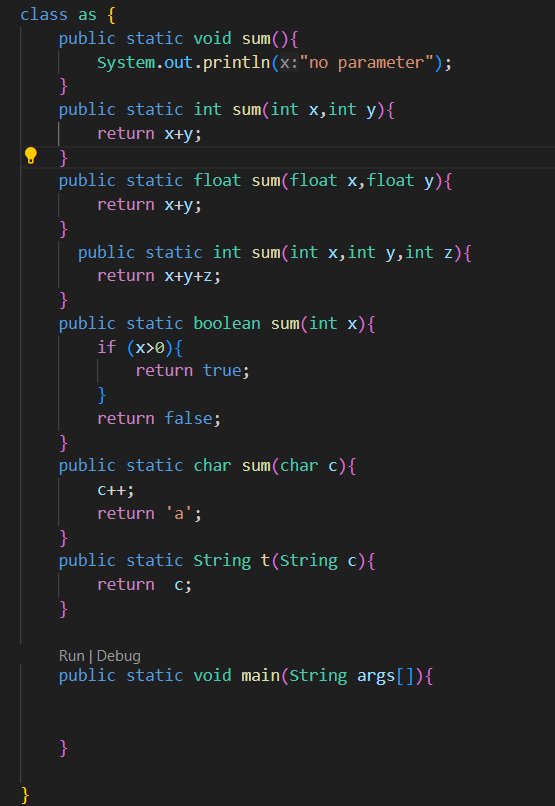


# 2)Entry Controlled Loop/pre tested loop (for loop, while loop)



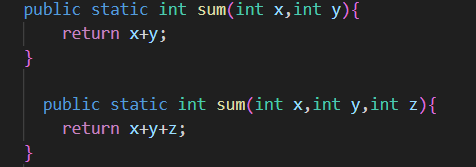


Function

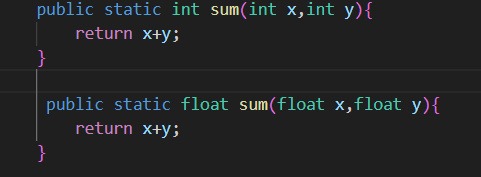
Here value is passed in function by call by value 

# Function overloading:

1. Parameter (same name different parameter)



1. Datatype (same name but parameter datatype different)

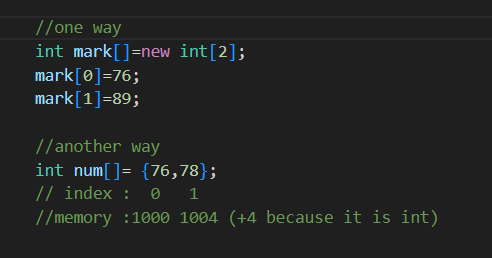


Array

*For understanding only*

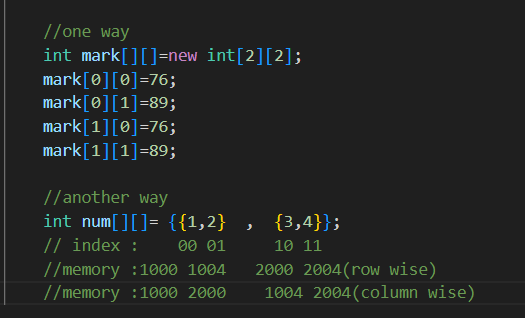
1. *Int dim[row]*
2. *Int dim2[row][column]*
3. *Int dim3[depth][row][column] (generally not used but can be)*

# One dimensional Array



# Multidimensional Array

2d and 3d array and many more

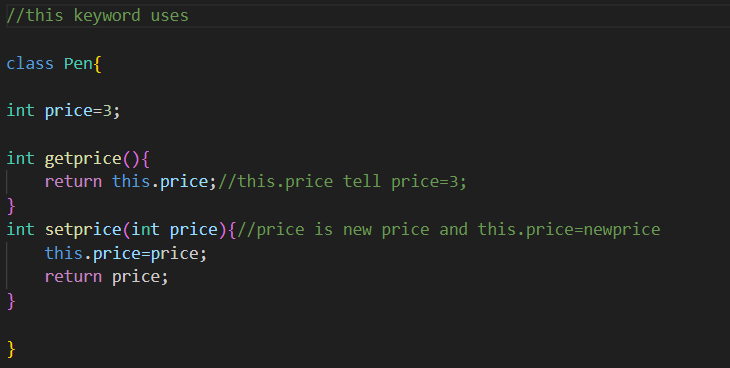


OOPS: Class

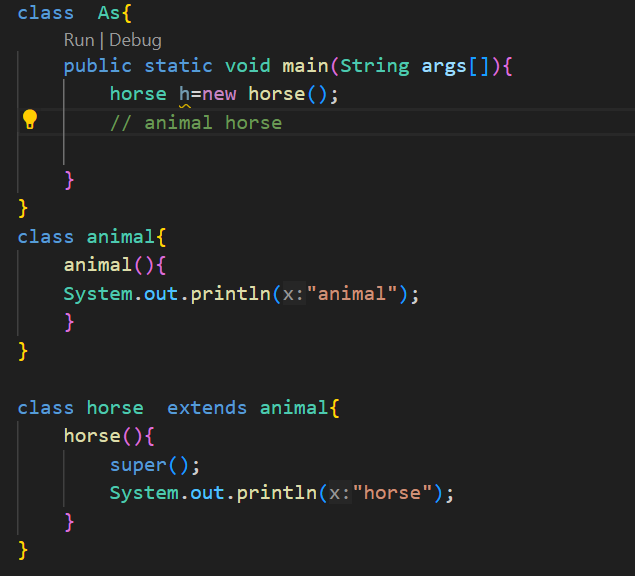
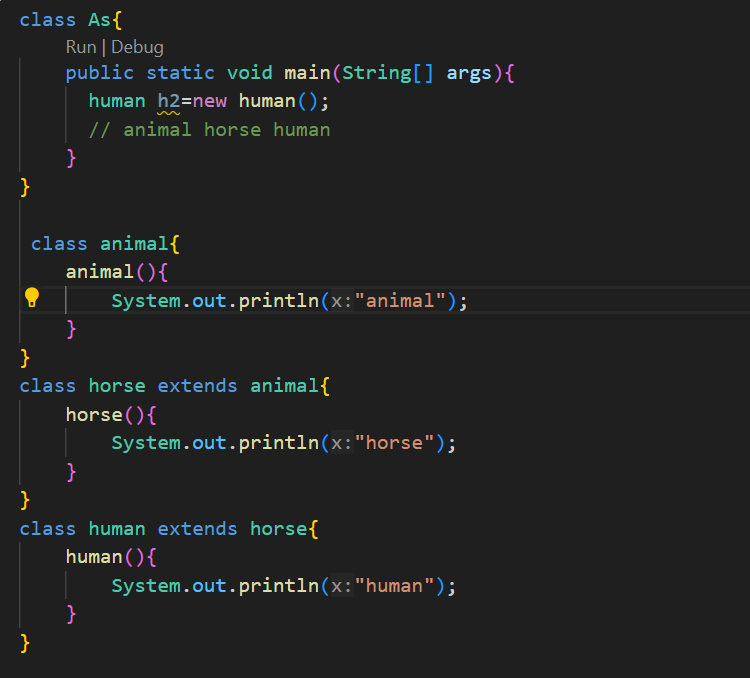
* Object: Entities in the real-world (ex-p)
* Classes: Group of these entities (ex-pen)
* this keyword: It is used to use already define variable(ex-this.price)

Note : By convention class first letter of name is written in capital letter

This keyword



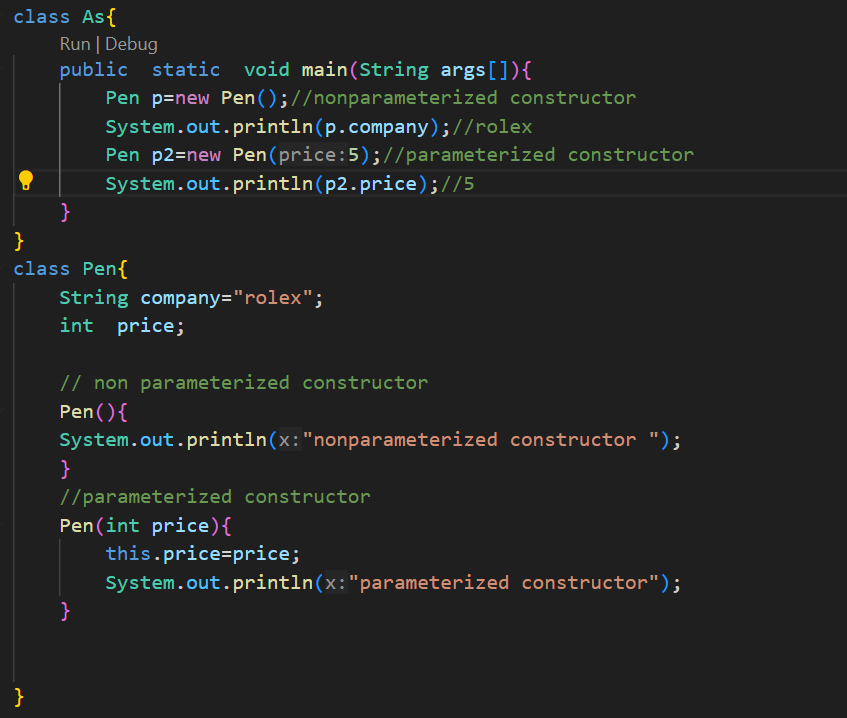
# How constructor is calling (super keyword is used by java by default if not used)



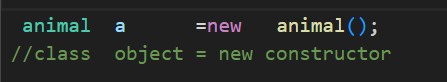
Static keyword: used to share the same variable or method of a given class ex-static property, static function, static blocks, static nested (class class A { class B{ } } )

Super keyword: use to refer immediate parent class object

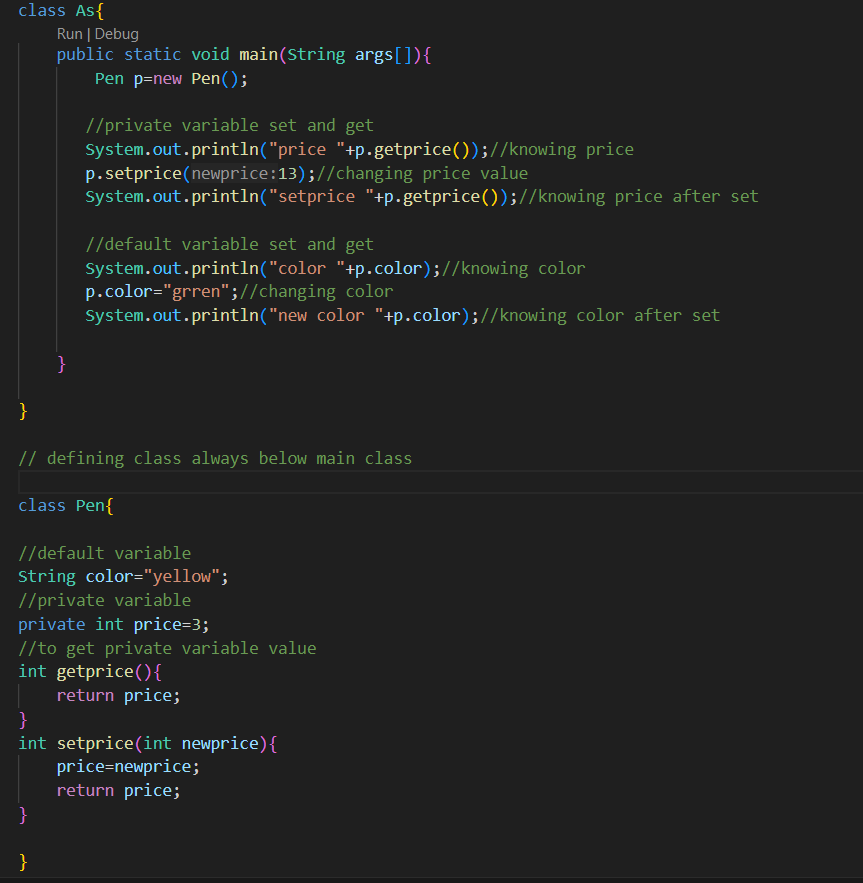
Constructor



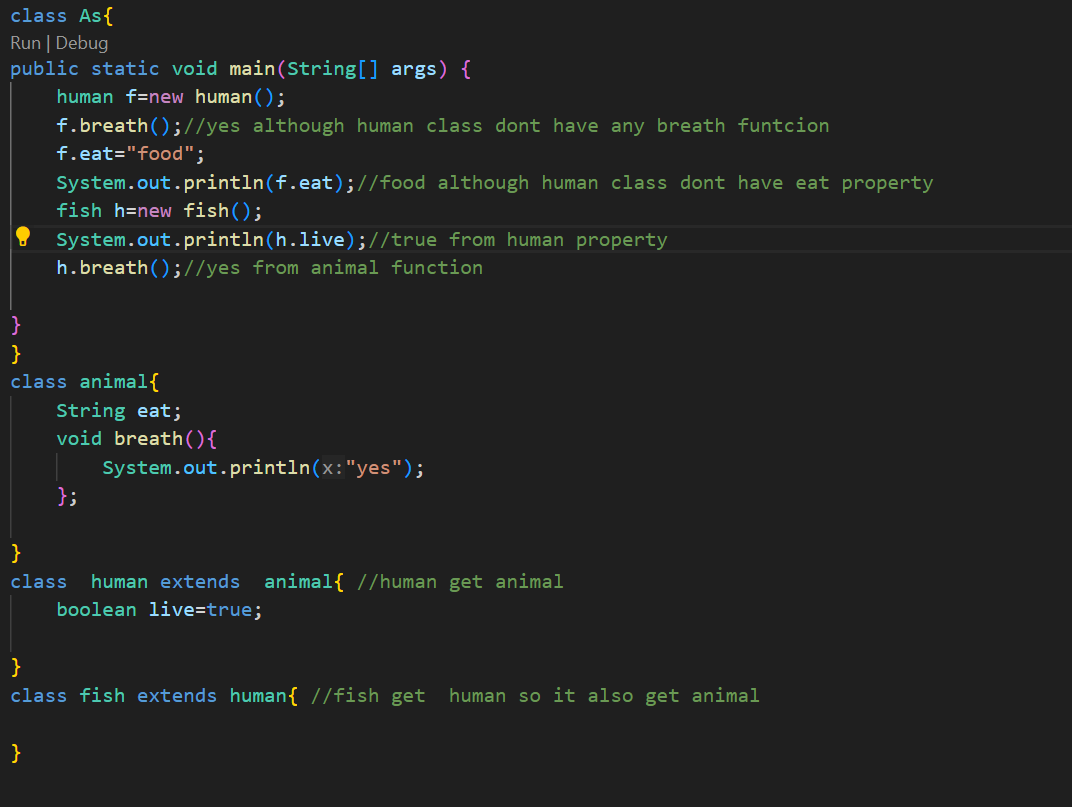
Class vs object vs constructor



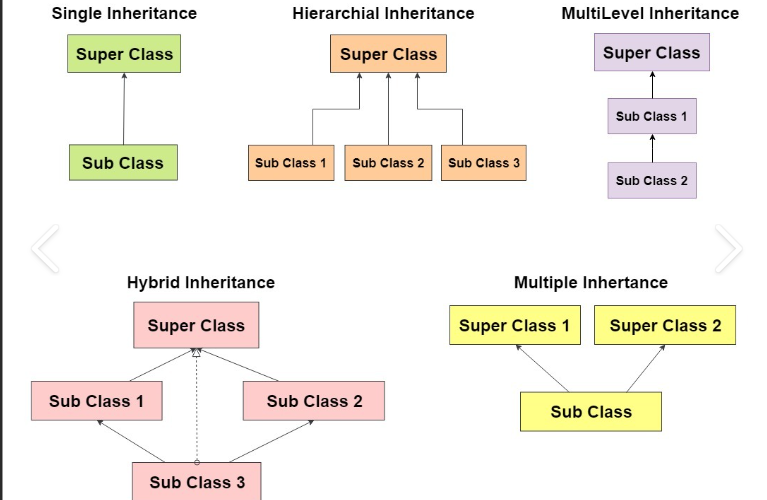
# Encapsulation (wrapping up data and methods under a single unit .it also implants data hiding using access specifier like default private protected)

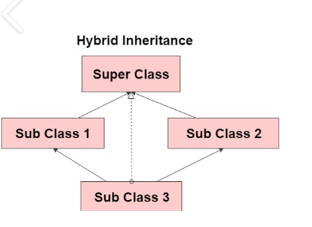


# Inheritance (passing of property and methods from parent class (base class) to child class (derived class))

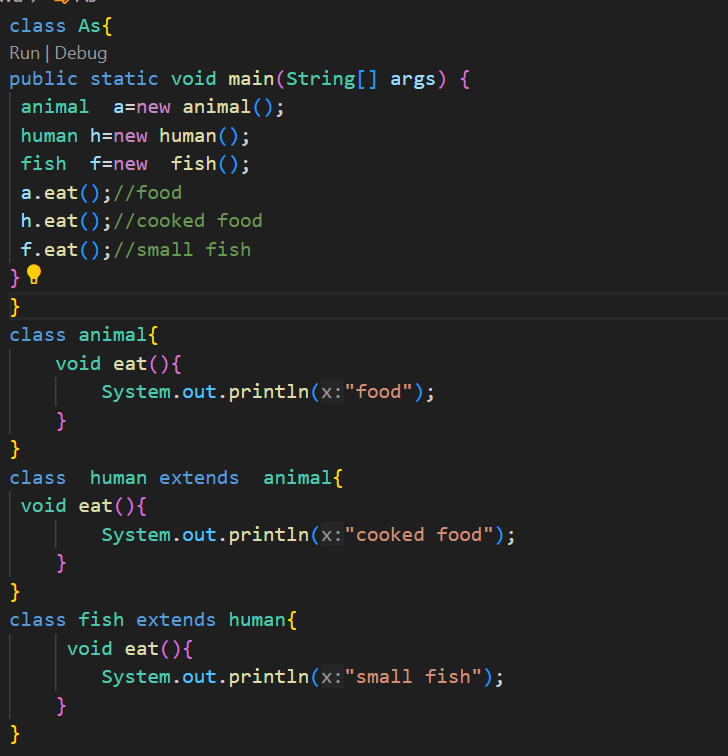


### Type of inheritance same logic will work as human extends animal

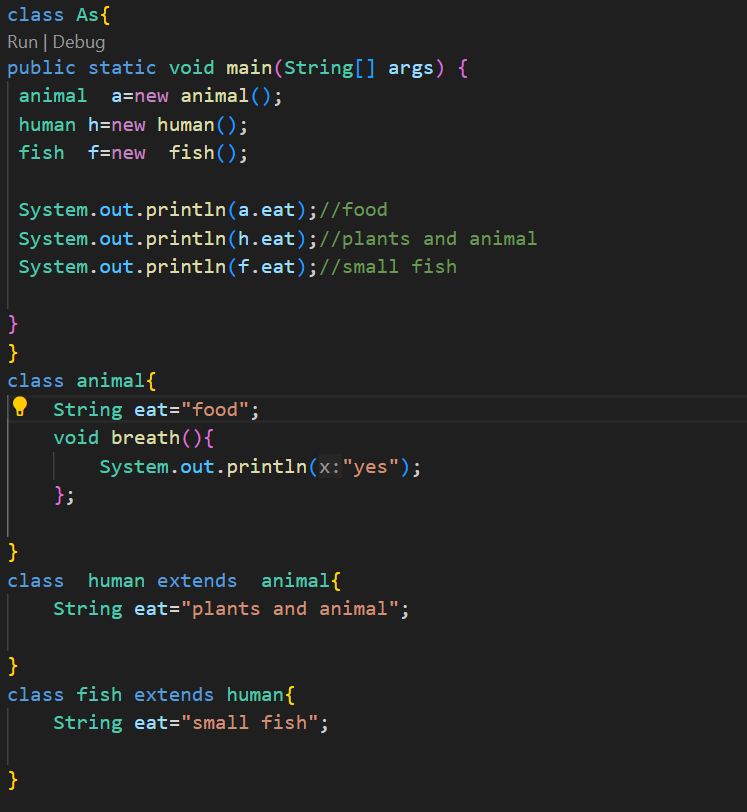




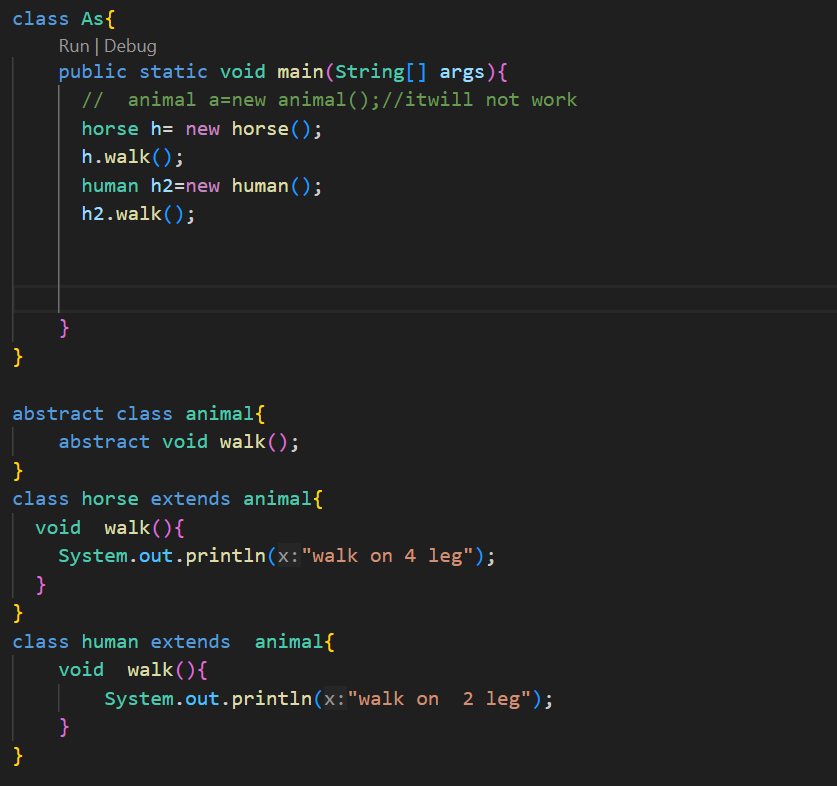
# Runtime/dynamic polymorphism (many forms for function only)



Variable shadowing

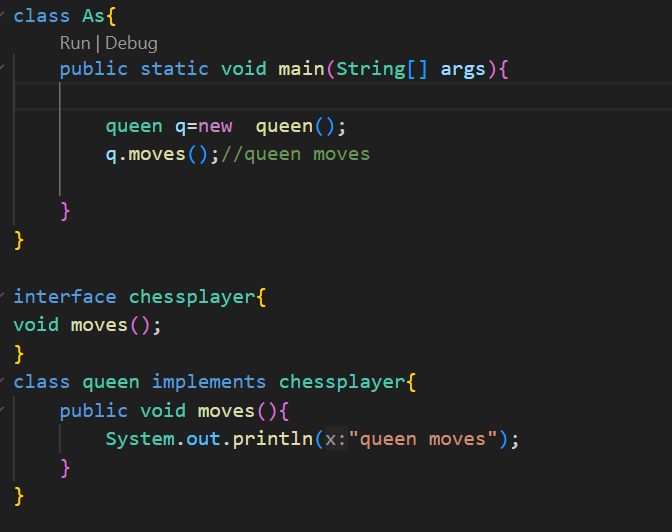
Same property in each parent and child class and its child classes 

# Abstraction (hiding all unnecessary details and showing only important parts while encapsulation hide important data like password)

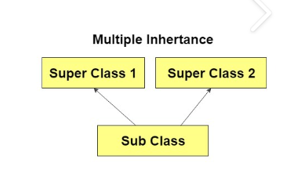


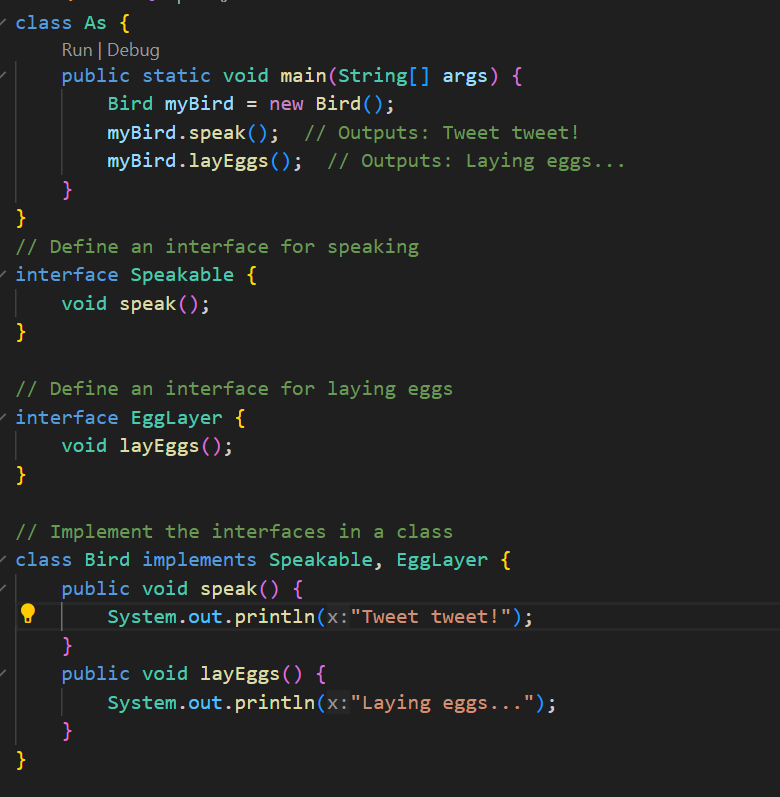
# 5.Interfaces (part of abstraction)

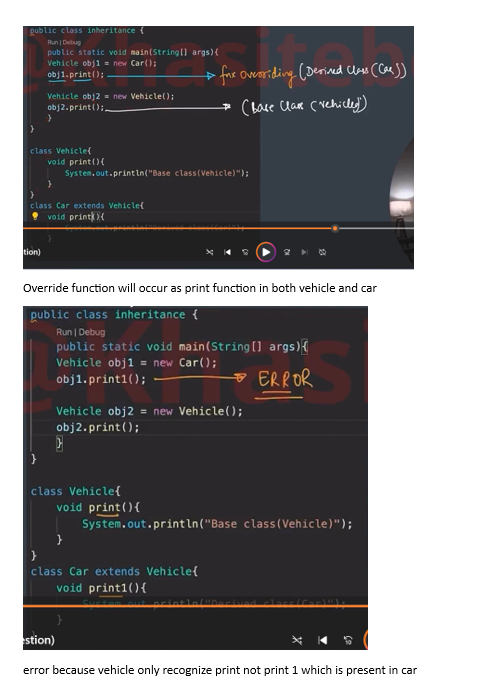
Total abstraction using interface



Multiple inheritance is not supported by java but can be implement using interface

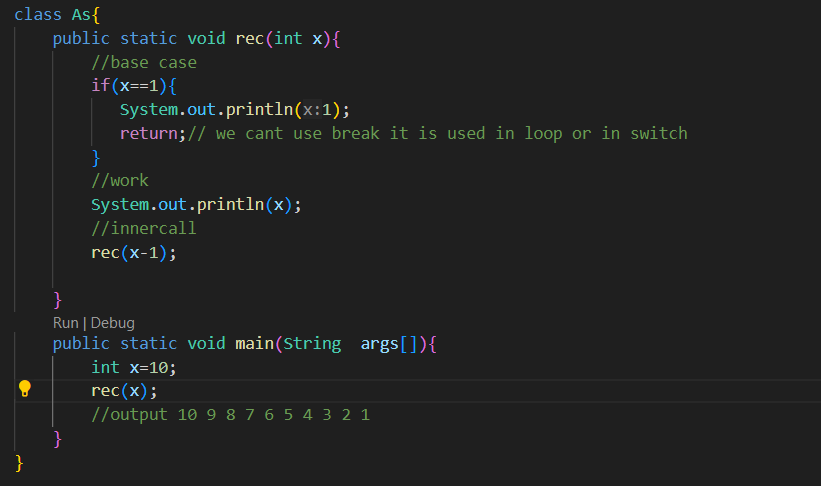






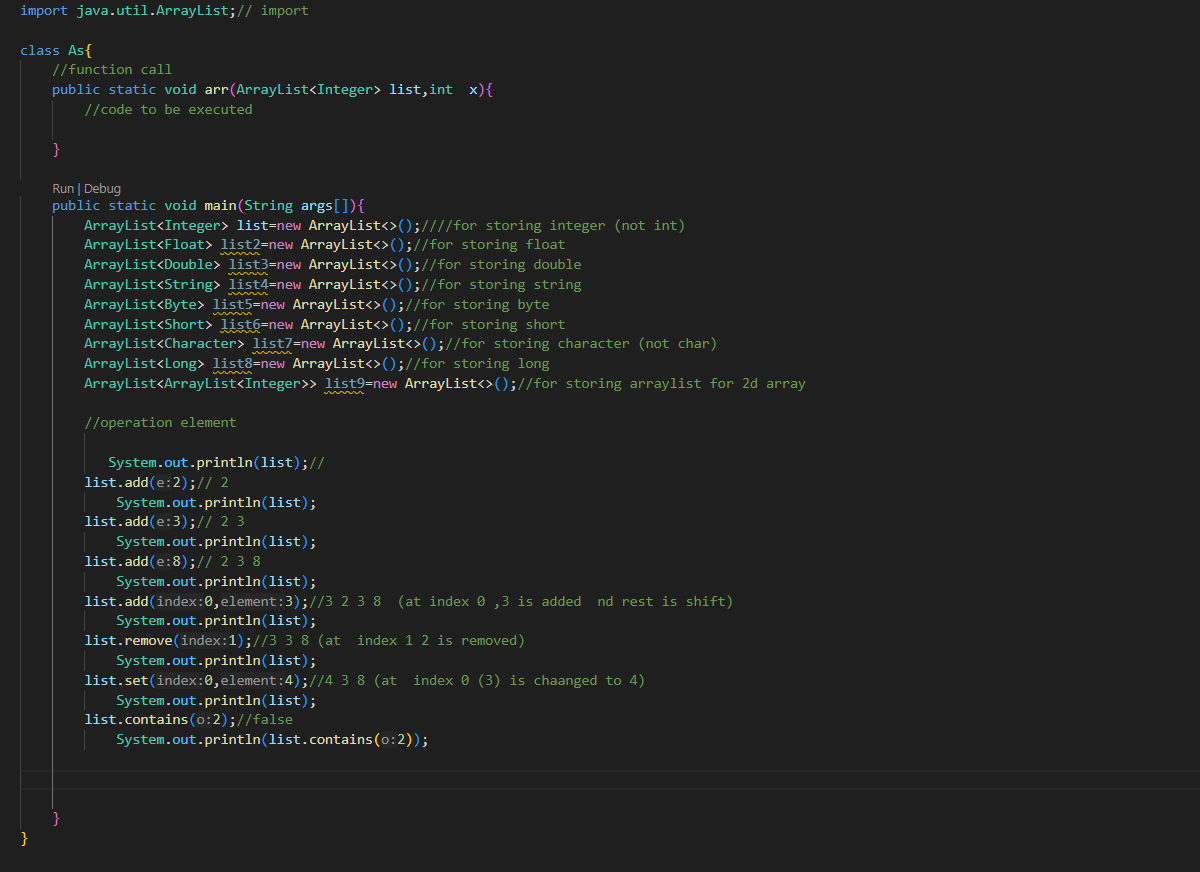
Recursion

1. Base case
2. Work
3. Inner case

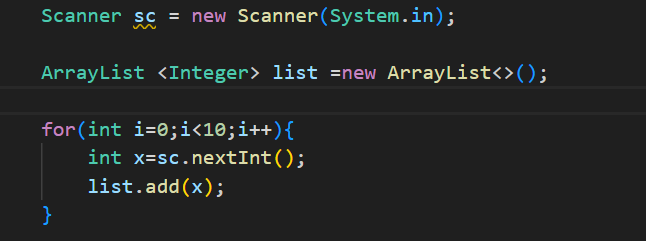


Array list

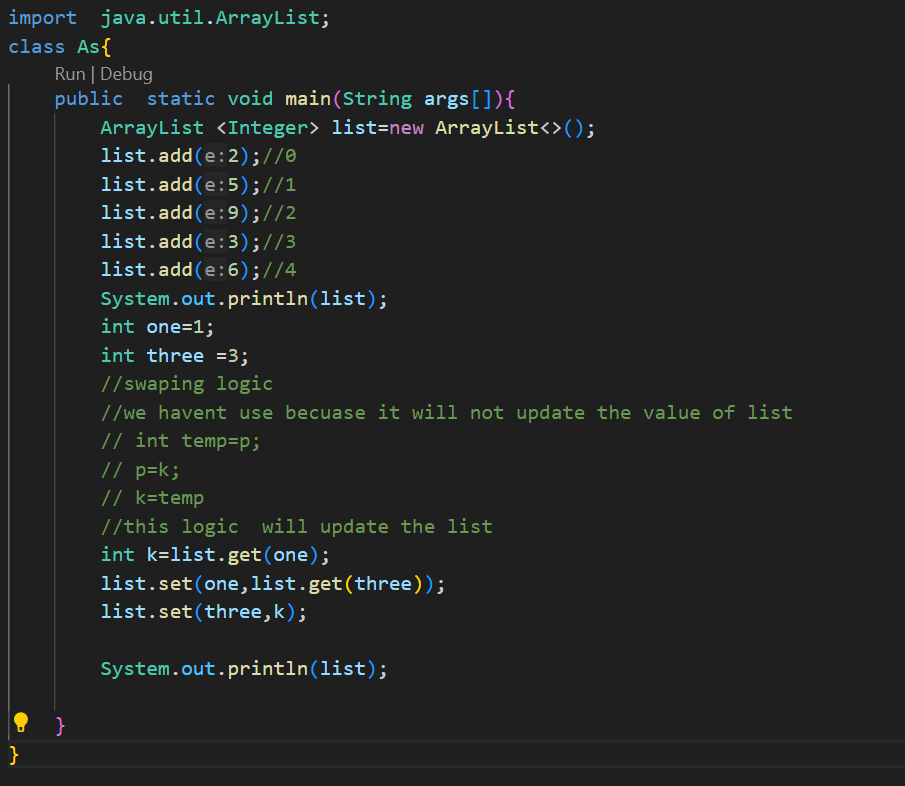
* similar to array but with dynamic size where primitive type of data like int can’t be stored directly. When we need extra space to store vriable ,in array list then it double the size



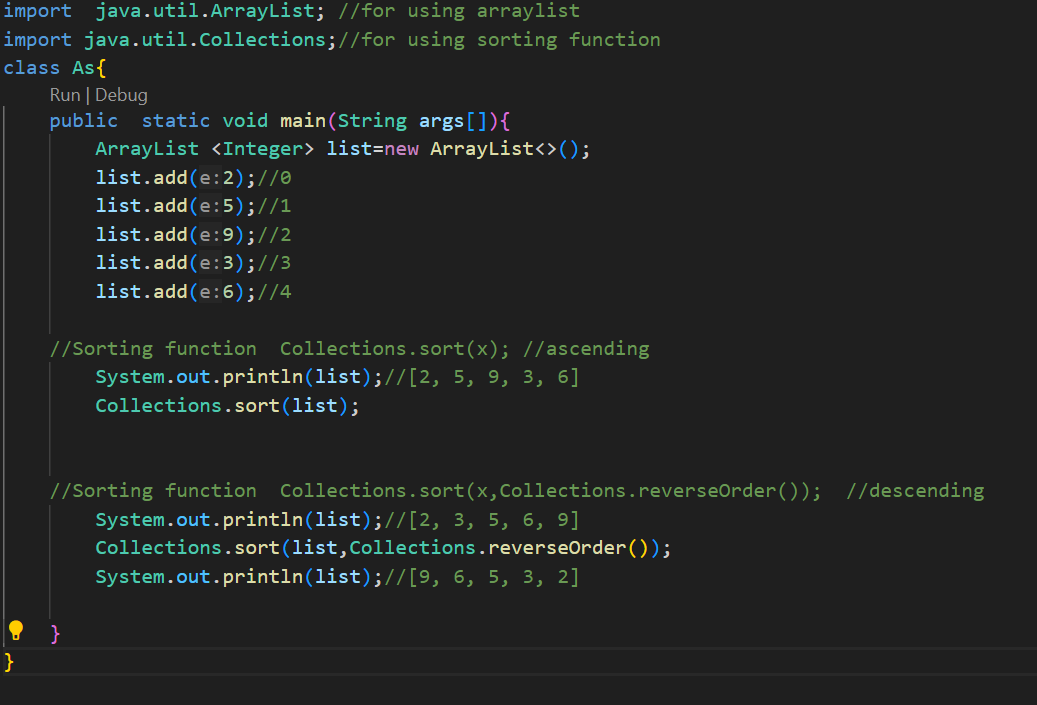
# Input using scanner class



# Swapping logic (for updating list)

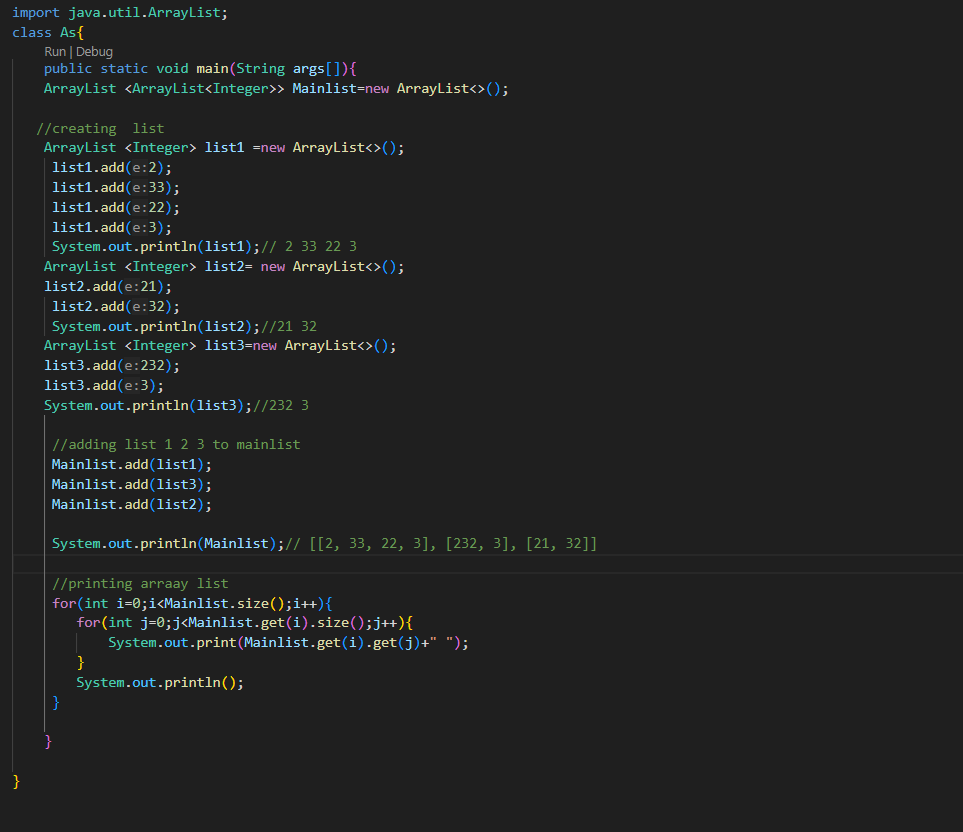


# Array list Function

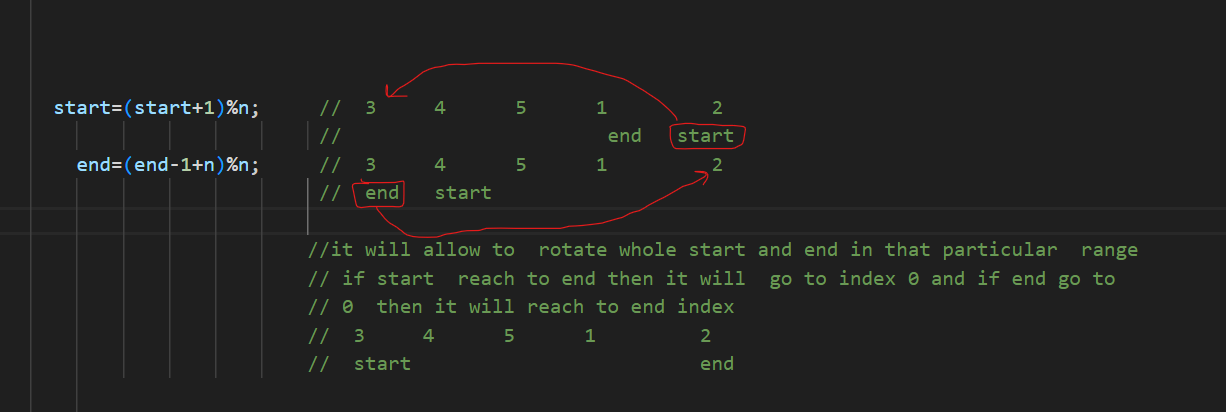


# Multidimensional Array List

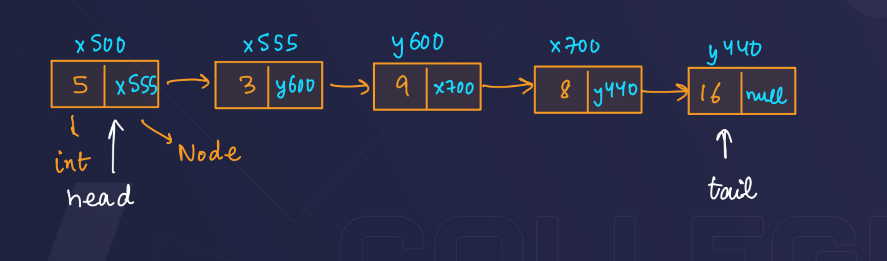
Ex-2d array list



# Modulus function property



LinkedList (Collection of nodes)

 In function public static void display (Node x) {}

