**Class 11 object and class topic date: 04-03-2024**

**Static is not polymorphic, mean it has only one value share across all the objects, it does not have separate values for each object. This mean when we create the object of a subclass then static method will have same values for all the objects of subclasses, that’s why this behavior is not inheritance. That’s why this not participate in inheritance.**

**That’s why static method does not overriding, not interface, not abstract.**

**What is oops?????**

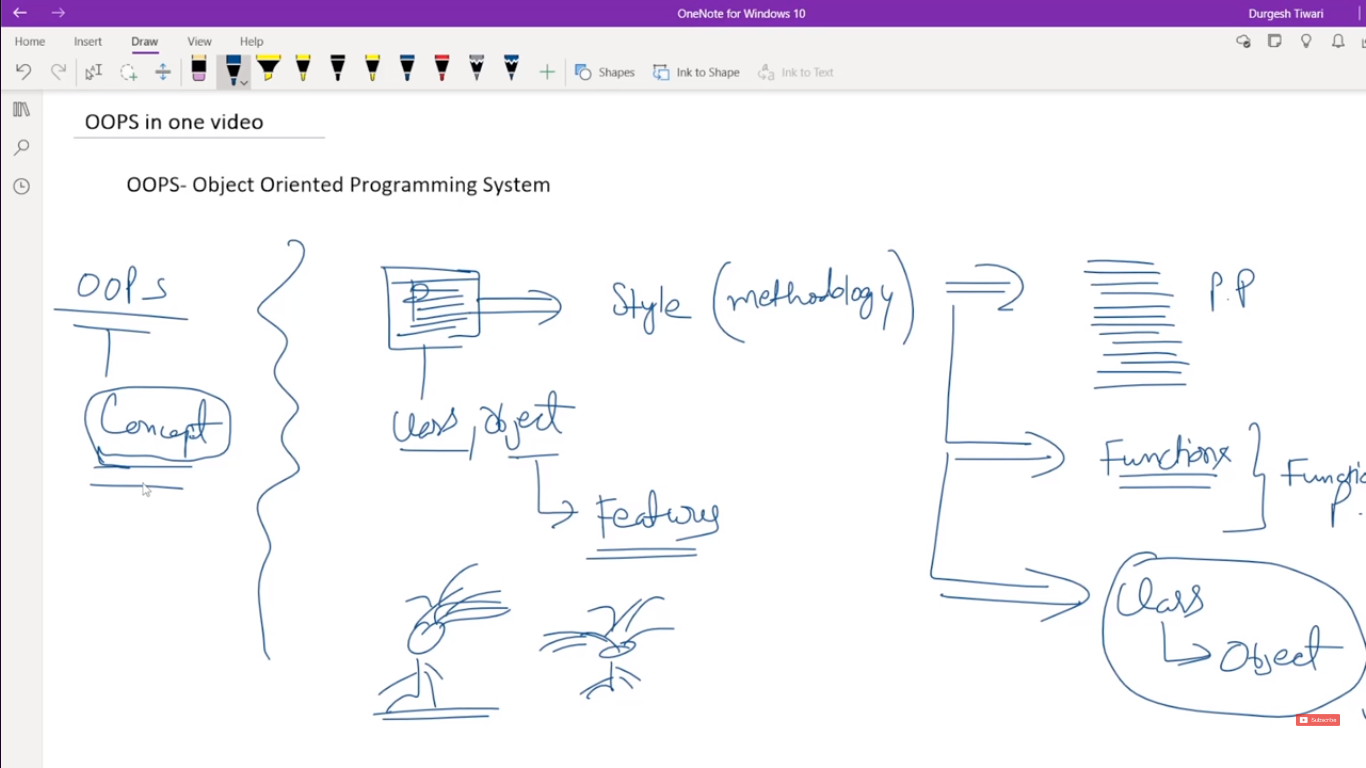
Suppose you say to a person please write a code; this is his own style how he writes the code.

**Oops is a style of writing code, is a methodology of code writing, is a set of rules, set of guidelines to write a code.**

1. if a person writes procedure step by step this is called procedural programming

2. if a person writes a code using functions this is called functional programming

3. if a person write code using class and object this is called oops programming.



**What is class and object???**

**Class; is a blueprint, dye, sketch plan, idea to create an object.**

suppose a person want to build a car, he directly cannot build a car , first he take hammer, iron, spare parts and then will make the car. This will not happen, first he has to plan what and how he will build the car or any other thing.

is a logical entity. Plan in your mind, or drawing on paper, not touch able.

Class is plan, idea, paper work to design anything.

Class will have data/properties/variables/attributes

Class will also have behavior/what thing will behave how.

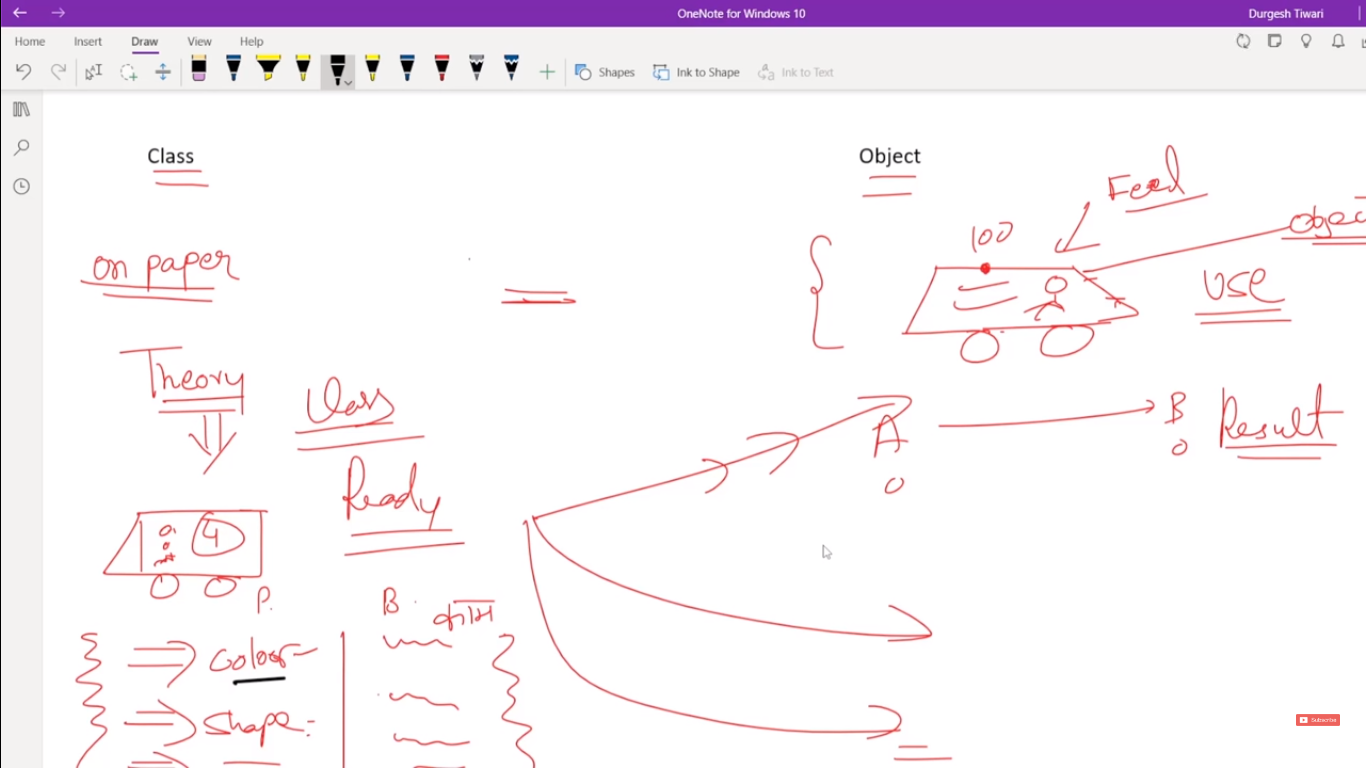
**Object**

an object is an instance of a class. object is the physical implementation of class. . Instance mean hawala(urdu hawala ,missal, namona )

object is the physical construction of a car. Object is also called instance of a class.

Object will have all the properties and behaviors from the class.

Creating an instance of a class mean creating object of that class.



If there is no import inside the package then java will check that class

**void** return type means that it doesn't return any value. It simply performs some action or task, but doesn't produce a result that can be used in further computations or assignments.

**System.out.println()** is a non-return method, this is a void method.

===============================================================================================================================================================================================================================================================================================================================

**CLASS 12 JAVA TOPIC STRING METHODS**

**What is a method???** Collection of statement , grouped to gather to perform task or operation

**What is mean calling a method??**

Calling a method in Java means invoking or executing the code that is defined within that method.

Types of methods???

Return type

And non return type

---------------------------------------------

Built in method

User defined method

---------------------------------------------------

Method with parameter

Method without parameter

-------------------------------------------------

If a method is return-type then its value can be stored inside the variable. And can be used for further calculation.

----------------------------------------------------------

**What is method signature??** Name of method and parameter types.

**What is method body??** The method body contains a collection of statements that define what the method does.

**What is method header??**

The method header in Java consists of the method's signature along with any modifiers.

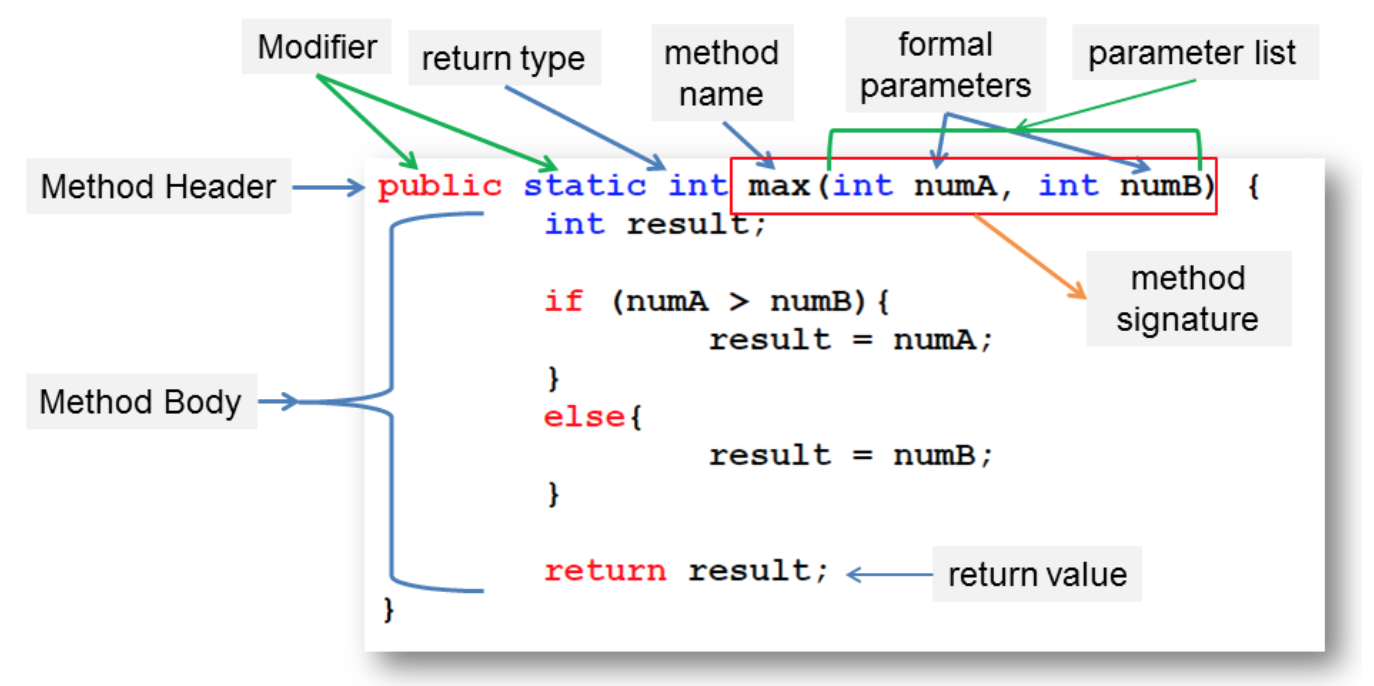
Parameters are enclosed within parentheses **()**, and if a method does not accept any parameters, the parentheses will be empty.

**What is a parameter???** define a variable inside the method.

**What is argument??** Assigning the value to the variables is present in signature called argument.

**What is parametrized method??** a method that contains variable in its signature called parameterized method.

**What is non parametrized method???** the method which do not contains variable in its signature.



**What is a prime number??**

Prime numbers are natural numbers greater than 1 that have no divisors other than 1 and themselves.

---------------------------------------------------------------------------------------

================================================================================================================================================================================================================================================================================================================================**JAVA CLASS NUMBER 13 STRING METHODS**

**What is string???** is a class designed to store alphabets(A,B,C) symbol@#$) numbers(123) in the form of group.

It is a data type used to represent text rather than numeric data.

String are objects. every string that you create is actually an object of type String class.

**What is data type???** which type of value a variable can hold and what type of operations can be performed on it.

**What are String methods???** In Java, the **String** class provides a wide range of methods to manipulate and work with strings. Here are some of the commonly used methods of the **String** class:

1. **Str.charAt(4);** this method tell at which index which character is present String str=”Batch 11 is awesome”;
2. **toCharArray();** this method converts string into an array of char. Char [] val=str.toCharArray();
3. **length method**; gives total number of character inside the string or array or array list.
4. **Length()-1** it will print last character from an array , because it starts from zero.
5. **String toLowerCase** method converts into small alphabets
6. **Strong toUpperCase** converts the string into upper case letter
7. **indexOf(“B”)** of method tell the index number of any character.
8. **Sub-String method**. It gives the substring/small string of original string. If you will put two value (0, 5) it will print only up to 5 characters, but if you will put (5) it will print all values from 5 to last character. This is very important.
9. **Replace method;** it takes two arguments, first the string which you want to replace, second the word which will be replace old word with new word.
10. **Trim method** will remove spaces before and after of the string.

**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**Regex methods;**

**Replace all method;** is used to replace each part of the string that matches with a specific pattern according to the given regular expression.

1. **Split(); method. is used to convert string into and array of string.split the string into many strings and store them inside the array.**
2. **Arrays.toString method is used to convert an array into a string.**
3. int[] array = {1, 2, 3, 4, 5};
4. String string = Arrays.toString(array);
5. System.out.println(string);

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**What is difference in replace and replace all method??**

**Replace method just replace old string with new string**

**Replace all method replace all string with matching pattern.**

**What is regex regular expression???** It's a sequence of characters that define a search pattern. regex is used for matching the patterns in strings.

A search pattern is like a secret code that helps you find specific pieces of information within a larger text.

**What is method chaining??** When multiple string methods are called/written in one line this is called chaining this is only possible in strings.

**What is concatenation???** **Concatenation**: combining two or more strings into a single string. This is typically done using the concatenation operator **+**

Increment operator.

**Count++; mean is ------ count = count + 1;**

**!=** is used for inequality comparison,

while **==** is used for equality comparison in Java.

===============================================================================================================================================================================================================================================================================================================================================================================================================

**JAVA CLASS 14 REGULAR EXPRESSION STRING BUILDER**

**String methods**

13. **starts with method**; it return Boolean , if a string starts with first letter.

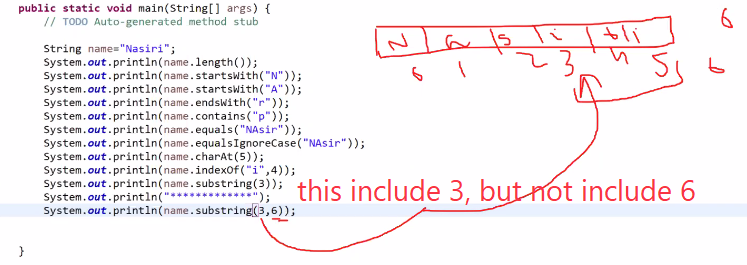
14. **ends with method**; it gives Boolean

15. **contains method**. it tells that a string contains a character or not.

16.**Equals method**; it will check string is equal or not case sensitive.

17.**equalsIgnorecase**; it will ignore the upper and lower case of string

18.String val=**ArraytoString(str); this method convert an arrays into string. You can print all elements from an array without using any loop.**



**To check the performance of any software, check two things??**

1. how much memory it taking

2. how much time it taking to process your request.

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

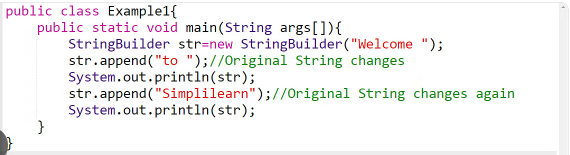
String and string object are same, when you create a string actually you create object of string class.

**How string works internally, how strings are immutable/unchangeable/unmodifiable???**

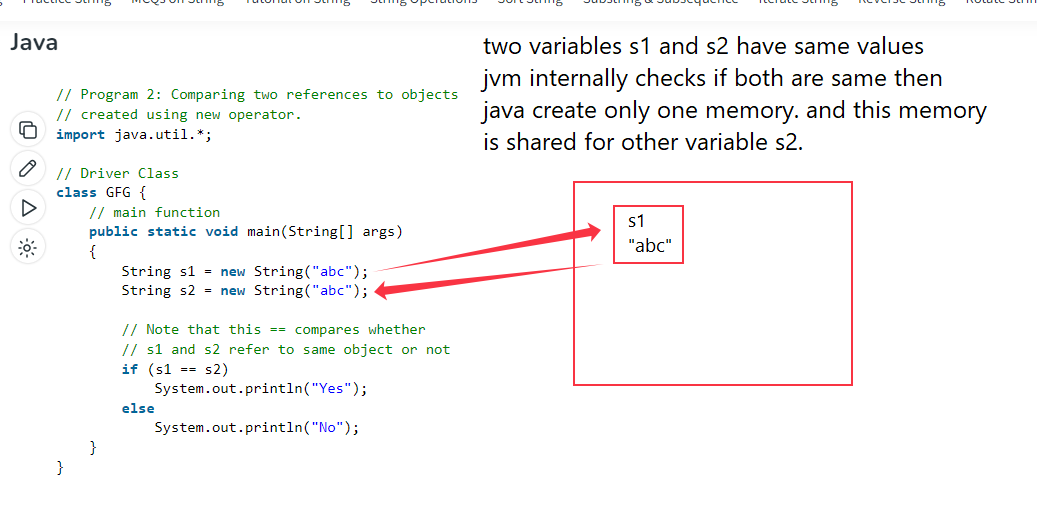
Once a string object is created its data or state can't be changed internally in memory. but if you have different values in string then a new memory for string object is created for new data.

**What is string builder??** The StringBuilder class in java is same as String class except it is mutable i.e. it can be changed.

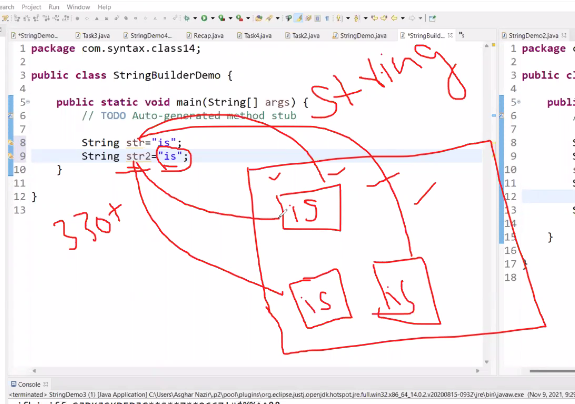
change able memory for each variable.



Whenever we create a new string object or string variable, JVM checks for the presence of the object in the memory, If String is available in the memory, then same object reference is shared with the variable, else a new object is created.







**When we have to use strings??**

1. when you need to save memory.

2. but its performance will slow, searching time

**When you want to use String builder???**

1. when you want to need performance , java will not search that already a variable present or not.

2. new memory will be created for every variable a lot of memory will consume.

String Builder class methods are very important.

1**. reverse method();**

**2.insert method(index 5, “hello”);**

**3. delete method (index 4, 6)** --- at index 4 which it will be deleted, 6 will not include here

==============================================================================================================================================================================================================================================================================================================================

**JAVA CLASS 15 TOPIC ; TYPES OF VARIABLE**

**How many types of variables are there??**

**1. what are instance variable----** **They are accessed using the reference variable to the object.**

**2. what are local variable – access modifier can not be used with local variables. Nor static.**

**3. what are static variable -- class level variables because they don’t belong to objects.**

Static variables and methods are accessed using the class name itself.

**Static variable can-not create inside the method**

**Non static variable can-not be called inside static method.**

Static variables used when you share the values across all objects

Instance variables are used when you share the values across all the methods.

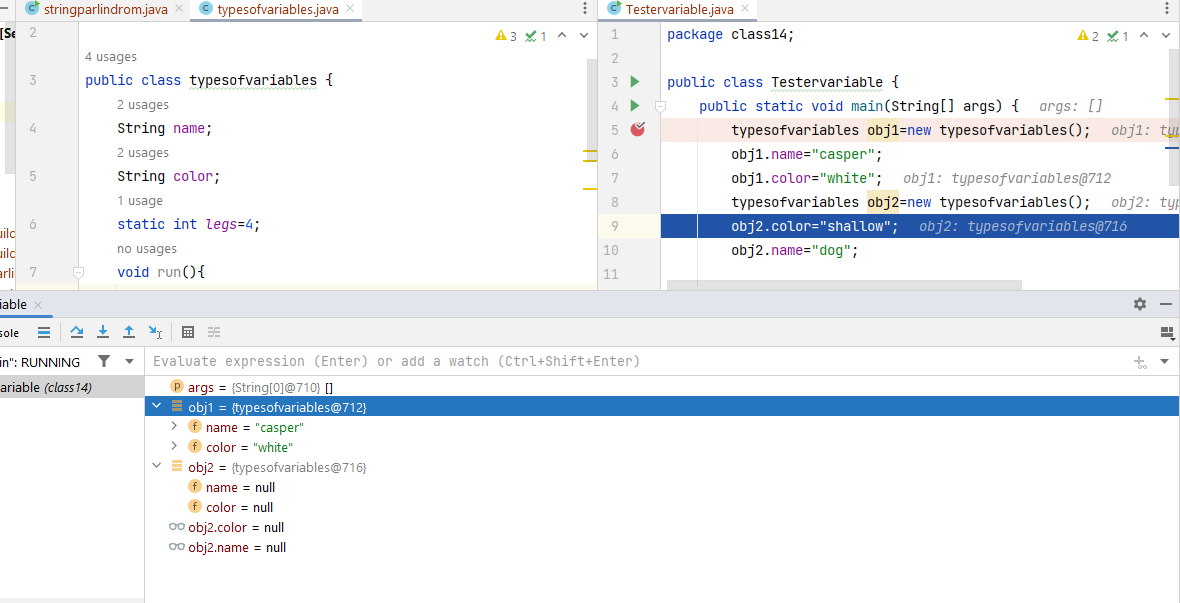
**When we should use instance variables**?? When you want share one thing in many methods in a class.

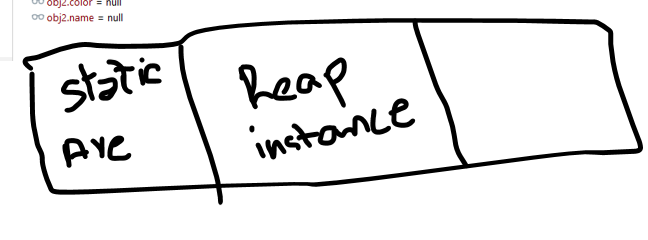
**Why we should not use instance variable**?? Because they occupy memory even after execution.

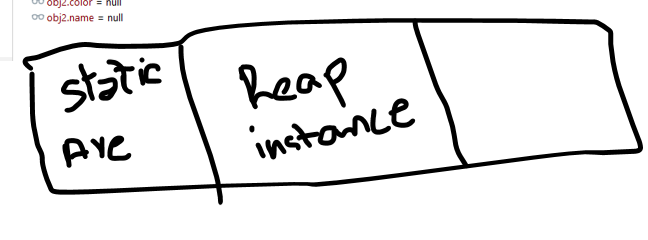
**When you use local variables??** Always use inside the method**.**

**For instance variables a separate copy is created for each object inside the heap memory.**

**But for static there is no separate copy is created. just one memory created and it shared for all the objects and methods.**

****





================================================================================================================================================================================================================================================================================================================================

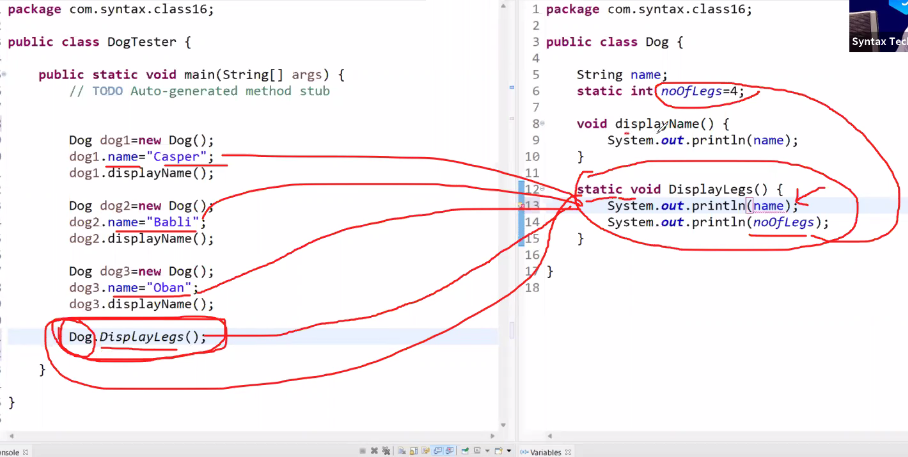
**JAVA CLASS 16**

Declare a variable int age;

Defining a variable and initialize int age=23;

Initializing the variable int age=0;

Assigning a value age=20;



Instance variables and instance methods mean belongs to objects,

Class level variables and methods mean belong to class. Can be called only with class name.

**Why instance variable can-not be called inside the static method???**

Because java is confused each object has different value for instance variable which value I have to use.

**Why non static method can-not be called inside the static method??**

Because each instance method will having must instance variables , and instance variables can-not be called inside the static method, that’s why non static method can-not be called inside the static method.

---------------------------------------------------------------------------

**Scenario; instance variables belongs to object , each object has separate copy of instance variables, if there are thousands of objects of a class, each object will have different value of instance variable, when you will call an instance variable inside the static method, java will confused from which object I have to take value of instance variable because there are thousands of objects and there values.**

Instance methods mean they belongs with object. You can only call then just by creating object of a class. That’s why they belong with object.

Java knows static variable has only one copy, it can call inside any method. in this case java will not be confused.

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**What are access modifier??**

Access modifiers are keywords that are used to control the visibility/accessibility of variables, methods and constructors in a class. --- class can have only two access modifier public, default.

The four access modifiers in Java are

* private = member accessible only within the same class in which private created. Not even subclass
* default = member it is accessible only within the same package, not accessible in another package.
* protected = member is accessible for sub classes, present inside the package even outside the package for inherited classes only. This is intermediate access, public is universal access. Some times you do not want to give access to all the members , protected is intermediate access.
* public =member is accessible inside all the packages present inside the project.

**What are no- access modifiers??**

Non-access modifiers provide information about the characteristics of a class, method, or variable to the JVM.

Seven types of non-Access modifiers are present in Java. We will study only 3.

1. [static](https://www.geeksforgeeks.org/static-keyword-java/)
2. [final](https://www.geeksforgeeks.org/final-keyword-java/)
3. abstract
4. [synchronized](https://www.geeksforgeeks.org/synchronized-in-java/)
5. [transient](https://www.geeksforgeeks.org/transient-keyword-java/)
6. [volatile](https://www.geeksforgeeks.org/volatile-keyword-in-java/)
7. [native](https://www.geeksforgeeks.org/native-keyword-java/)

**Class and Interface cannot be declared as private**

If a class has private constructor, then you cannot create the object of that class from outside of the class

==============================================================================================================================================================================================================================================================================================================================================================================================================

**JAVA CLASS TOPIC – CONSTRUCTORS IN JAVA. 08-03-2024**

With methods always use public keyword,

With variables always use private keyword in practice;

**What is constructor???**

constructor is a special type of method which is used to initialize the object.

-----------------------------------------------------------------------------------------------------------------------

* Constructors have the same name as the class and do not have a return type, not even **void**.
* Constructors are automatically invoked when an object is created using the **new** keyword, while regular methods are called explicitly by name.
* Constructors have the same name as the class they belong to, while regular methods can have any name.

---------------------------------------------------------------------------------------------------------------------

**What is default constructor???**

The constructor which initializes the object's variables to their default values.

Default constructors are often called as "no-argument" or "zero-argument" constructors.

--------------------------------------------------------------------------------------------------------------------

**What is the purpose of constructor???**

If There is a class, where must object will be created to access the code inside the class.

And constructor control the access of class, if you want to create object you must initialize the variable first then you will access the whole class otherwise not.

Encapsulation. Constructor provide the control on class using.

----------------------------------------------------------------------------------------------------------

**To initialize an object there are many methods in java. We will use only one constructor.**

**-----------------------------------------------------------------------------------------------------------**

1. initialize by instance variables ---instance variable is assigned values inside the class

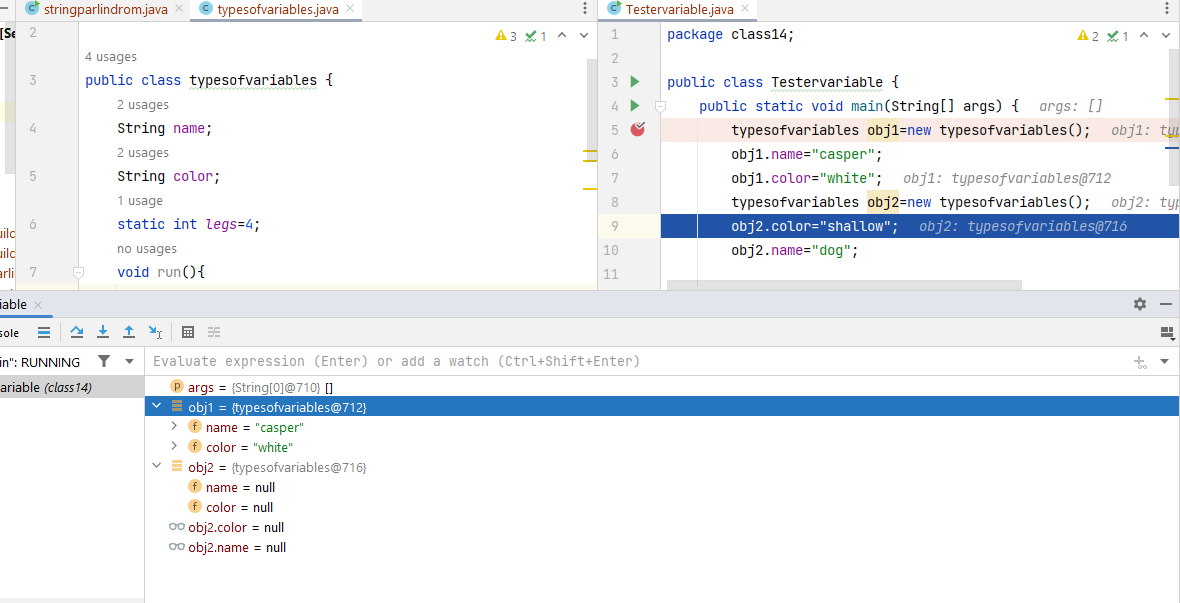
2. initialize by object itself-----instance variable is assigned value by creating object reference

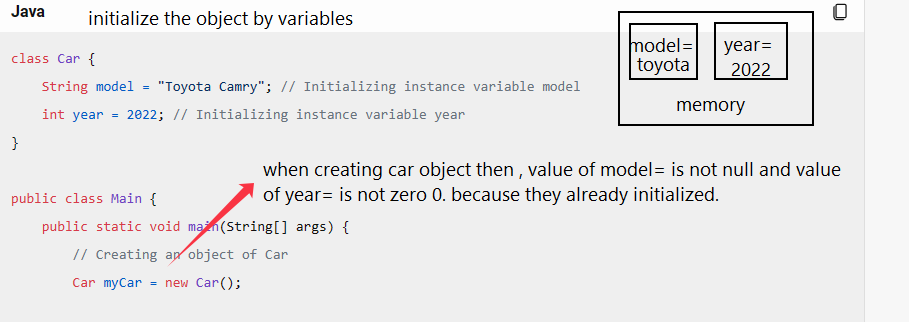
3.initialize by methods-----methods with parameters---- remember this.

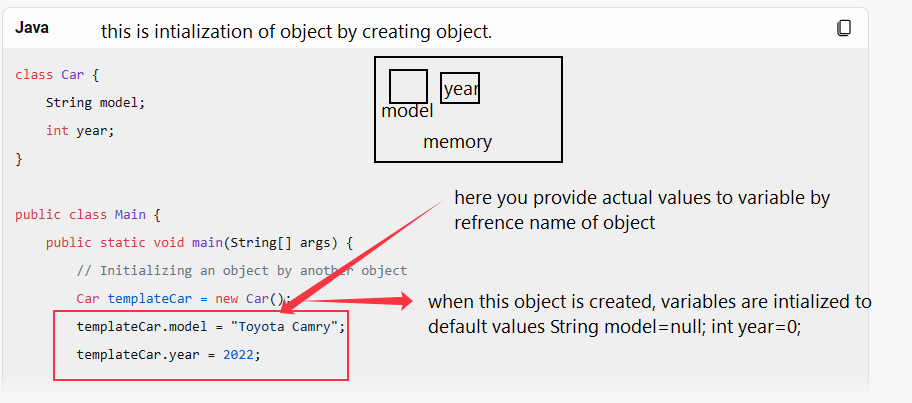
4. initialize by constructor.---by using constructor.

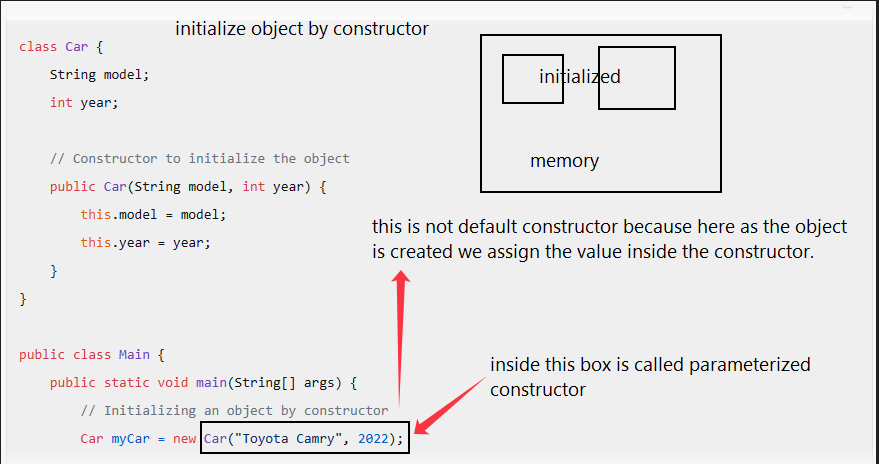
**The best practice to initialize**

object is by constructor only. others way occupy extra memory.

****







**If there is no constructor then what will happened???**

**Without constructors, we’d have to manually assign values after creating objects, which would be inefficient.**

**There will be waste of memory.**

**Definition: Declaring a variable means specifying its data type and name.**

**Defining and initialize are same.**

**Definition: Defining a variable means assigning a value to it for the first time.**

**What is initialize??** the process of assigning initial value to a variable is called initialize variable.

-----------------------------------------------------------------------------------------------------------

If you don't define any constructor in your class, Java provides a default constructor automatically. This default constructor initializes instance variables to their default values (e.g., **null** for reference types, **0** for numeric types, **false** for Boolean).

If you define any constructor (with arguments) in your class but don't define a no-arg constructor explicitly, Java won't provide a default constructor automatically. In this case, if you attempt to create an object using the default constructor syntax (e.g., **new MyClass()**), a compilation error will occur.

**If we don’t provide default constructor and just create constructor, why java gives error??**

This is because Java only provides the default constructor if you don't define any constructors at all. If you define any constructors explicitly,

Java expects you to use one of those constructors for object creation.

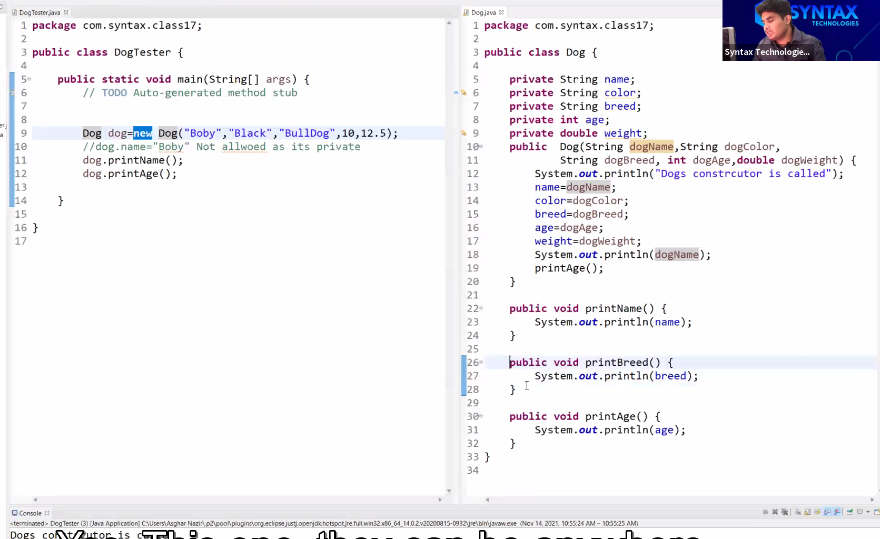
-----------------------------------------------------------------------------------------------

A method can be called inside the constructor

A constructor can be called inside the method.

Every action you can perform with constructor as you perform with methods.

-------------------------------------------------------------------------------------------



**Types of constructors??**

**1. What is parametrized constructor??** a method that contains variable in its signature called parameterized method.

**2. What is non parametrized constructor???** the method which do not contains variable in its signature.

**How generate constructor automatically??**

**Write click, select option, generate >>constructor**

**-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**“This” keyword is used to differentiate between instance variables and local variables that have the same name.**

**This(name, color); can be used to call other constructor present in same class.**

**------------------------------------------------------------------------------------------------------**

**-------------------------------------------------------------------------------------------------------------------------**

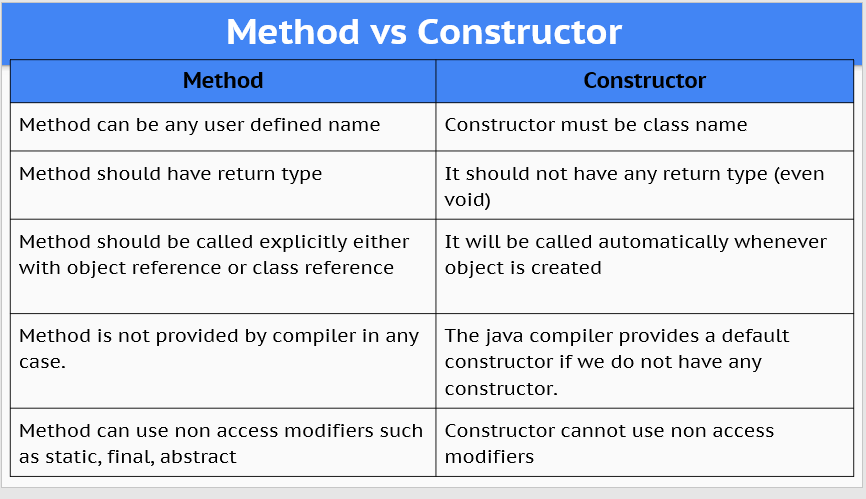
**This();**

**The constructors provide flexibility in object creation, allowing different combinations of parameters to be used for initialization.**

**This allows users to initialize objects of the class in various ways, depending on which attributes they want to specify during object creation.**

**Why constructor can not be static??**

**Because static method and static constructor can-not access the non-static variables.**

**Top of Form**

**If there are multiple constructors in class, which constructor will call???**

1. **Number of Parameters**: Java selects a constructor based on the number of parameters provided during object creation. If the number of parameters matches a constructor's parameter list, that constructor is chosen.
2. **Types of Parameters**: If multiple constructors have the same number of parameters, Java selects the constructor whose parameter types match the types of the arguments provided during object creation. It looks for the closest match in terms of parameter types.
3. **Order of Parameters**: If there are multiple constructors with the same number of parameters and matching parameter types, Java selects the constructor with the parameter types in the same order as the arguments provided during object creation.

================================================================================================================================================================================================================================================================================================================================

**CLASS 18 INHERITANCE ONE OF OOPS CONCEPT.**

**When multiple classes has common methods and fields, then there a concept of inheritance comes, you don’t want to write the code again and again and you extends the code from parent class and use it.**

**The main purpose of inheritance is reuse-ability of existing code. that you don’t need to write the code again and again, use a common code written already by developers and use this by inheritance.**

**----------------------------------------**

**Tasks Achieved by Inheritance but Not by Object Creation:**

**Tasks Achieved by Inheritance but Not by Object Creation:**

One main difference in object creation and inheritance is that , method overriding, polymorphism.

And more other differences are as follow.

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**What is inheritance oops concept in java???**

Inheritance in java is a process/mechanism in which subclass inherits all the properties and methods of its parent class.

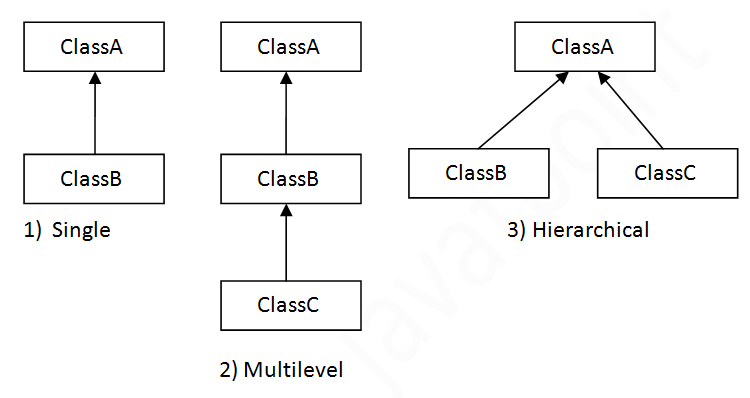
**Extends** is the keyword used to inherit the properties\methods of a parent class.

Extends keyword means that the subclass can reuse the code of the superclass and extend its functionality by adding new fields or methods, or by overriding existing ones.

**Types of Inheritance**

1. Single inheritance
2. Hierarchical inheritance
3. Multilevel inheritance
4. Multiple inheritance

In inheritance you create the object of only subclasses not super class.



**Advantages of inheritance???**

1. Application development time is less.
2. Application take less memory.
3. Application execution time is less.

**Disadvantages of inheritance???**

If any thing is changed in parent class, then all the child class will be affected. Tightly coupled classes. You have to make change in all child class also.

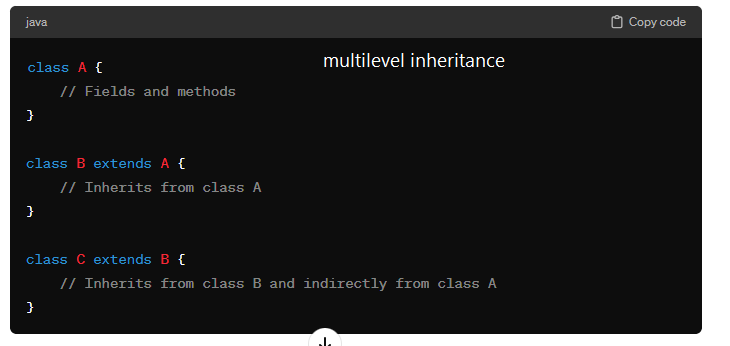
**Types of inheritance???**

Single inheritance ----- inheritance there exists single parent class and single child class.

Hierarchical inheritance---single parent class has many child classes.

Multilevel inheritance--Each subclass acts as the superclass for the next subclass in the chaining.

Multiple inheritance--- not allowed in java, one child class has multiple parent classes.





**What is “is-A” relation-ship??**

a subclass (derived class) is said to be a specialized version of its superclass

The **Dog** class inherits the **eat()** method from the **Animal** class, which demonstrates the "is-a" relationship.

**Which members don’t participate in inheritance??**

Private members and constructors don no participate in inheritance.

================================================================================================================================================================================================================================================================================================================================

**JAVA CLASS 19 SUPER KEYWORD 1 ramzan 12-03-2024**

**Why more then two class not be public??**

Because java will confused which public class I have to execute.

--------------------------------------------------------------------------------------------------------------

As you create the object of any class, constructor automatically will be executed.

public static void main(String[] args) {

C third=new C();  
 }  
}  
class A{  
 public A(){  
 System.*out*.println("A");  
 }  
}  
class B extends A{  
 public B(){  
 System.*out*.println("B");  
 }

-----------------------------------------------------------------------------------------------------

Constructor do not participate in inheritance.

That’s is why you can not write constructor in subclass.

--------------------------------------------------------------

**What is difference in this() keyword and super() keyword???**

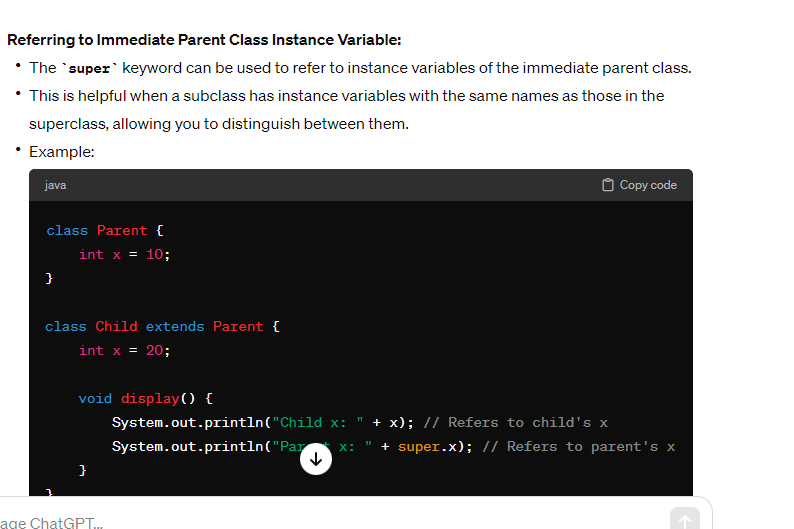
**this()** is used to invoke the constructor within the same class.

**super()** is used to invoke the constructor of the superclass inside any sub class.

* Super can be used to refer to the parent class instance variable.
* Super can be used to invoke the parent class method.
* Super() can be used to invoke the parent class constructor."

**Inheritance:**

* **super** can be used to refer to the parent class instance variable, invoke the parent class method, and invoke the parent class constructor.
* **this** cannot be used to refer to the superclass or invoke superclass methods or constructors. It is strictly used for referring to the current instance of the class.

Top of Form

----------------------------------------------------------------------------

**What is super keyword???**

Super keyword is used to call the constructor of parent class inside the child class. Because constructor does not participate in inheritance. That’s why we use super keyword.

------------------------------------------------------------------------------------------------------------------------

**Why constructor does not participate in inheritance???**

1. However, constructors are not members of a class in the same way that fields and methods are. Constructors are special methods with the same name as the class, and they are not inherited by subclasses.
2. **Constructor Is Not Inherited:** When a subclass is created, it inherits the members of its superclass, but it does not inherit the constructor of the superclass.

------------------------------------------------------------------------------------------------------------------

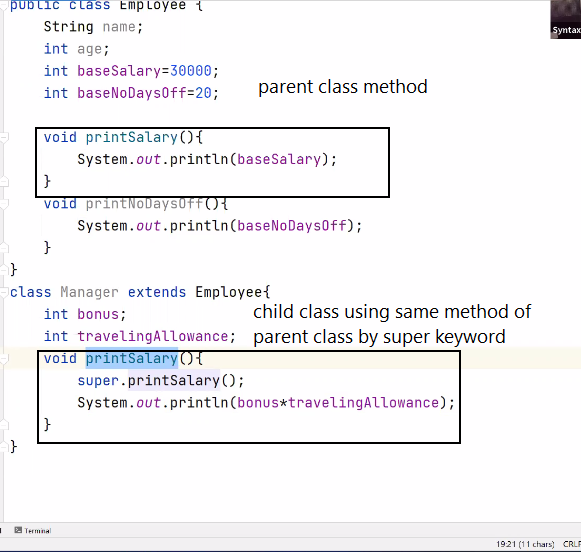
Instance variables are members of a class, but constructor are not member of a class like methods.

Their purpose is to initialize the objects state.

-----------------------------------------------------------------------------------------------------------------

Another use of super keyword.

If parent class has a method print salary and we want to use that method having same name in child class then you can use super keyword. Because you want to add some mor functionality to this method.



**Very important;**

Super keyword always called inside block of code of child class method or inside child class constructor only.

===============================================================================================================================================================================================================================================================================================================================================================================================================

**CLASS 20 , TOPIC OVERLOADING AND OVERRIDING.**

**What is method overloading??/static binding/early binding/compile time polymorphism**

In java there is a feature that allow a class to have multiple methods with same name but different number of parameters

multiple methods in a class having same name but with different number of parameter or different order of parameters or different types of parameters is known as method overloading

Method overloading is the process of defining multiple methods in the same class with the same name but different parameters.

----------------------------------------------------------------------------------------------------------------------

**What is var arg-s???**

variable-length arguments, is a method which accept N number of arguments of same data type.

This is particularly useful when you're not sure in advance how many arguments will be passed to a method.

the var args parameter behaves like an array of the specified type.

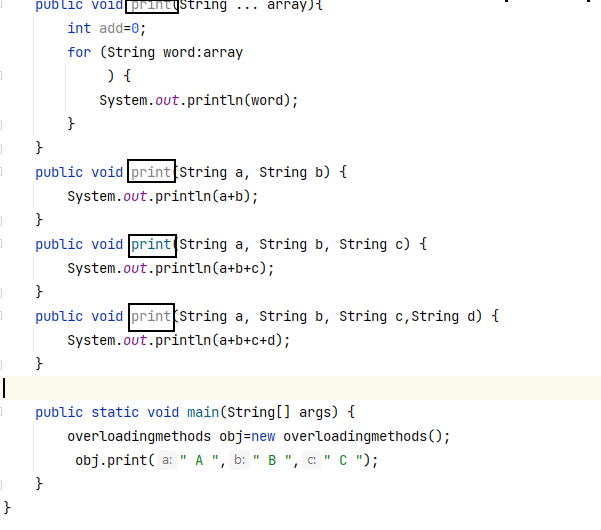
-------------------------------------------------------------------------------------------------------------------------

**Why we need var args???**

When you have multiple methods with same name and with same data type but different number of parameters, and you don’t need to want write methods again and again. Then there is a feature called var args that take n number of parameter having same data type.

--------------------------------------------------------------------------------------------------------------

In overloading no need of inheritance. the things which do not participate in inheritance also not participate in overriding.



**Only method signature has impact on method overloading.**

Access modifier, return type, non-access modifier have no impact on overloading.

Advantages of method overloading???why we do need ??

1. Code readability
2. Flexibility
3. Clean of code

**What is purpose of method overloading???**

When multiple method in a class perform similar task, and you don’t want to repeat methods again ang again.

This allows for the creation of methods that perform similar.

=============================================================================================================================================================================================================================================================================================================================

**Overriding**

When you override a method in a subclass, you're providing a new implementation specific to that subclass. This new implementation is independent of any other subclass or the superclass.

It provides a solution to the problem of superclass changes affecting subclasses by allowing subclasses to provide their own implementations of inherited methods.

**When we need overriding????**

**When you don’t like a method of parent class and want to make some changes in it then we use method overriding. By inheritance.**

کسی دوسری چیزکی نسبت زیادہ اہم

Gaalib hona .

**What is method overriding????**

When a method is existing in both parent and child class with same name, same number, type and order of parameter is known as Overriding.

This is used when you want to make some changes in overridden method child class.

**Non access modifier (final, static) don’t participate in method overriding.**

**Private access modifier don not override, because not accessible to subclasses.**

**What is the difference in method overriding and method overloading??**

1. **method signature**
2. **Method overriding occurs in inheritance classes**
3. **Method overloading can occur in any class**

**--------------------------------------------------------------------------------------------------------------------**

**Method Signature**:

* **Method Overriding**: the overridden method in the subclass must have the same method signature (name, return type, and parameter list) as the method in the superclass.
* **Method Overloading**: In method overloading, the overloaded methods must have different method signatures, typically by having a different number or types of parameters.

**What is the purpose of method overriding????**

When you don’t like a method in parent class and want to make some changes in it then we use method overriding. By inheritance.

**This allows a subclass to replace or modify the behavior of the method inherited from the superclass.**

**What is override annotation??**

**This is indication tells that you are correctly overriding a method or not.**

**Rules for Method Overriding**

* must be IS-A relationship (inheritance).
* method must have same name as in the parent class and child class.
* method must have same parameter as in the parent class.
* Method must have the same return type as in parent class

**------------------------------------------------------------------------------------------------------------------------------**

**The method signature (i.e. method name, parameter list) and return type have to match exactly in super and subclass.**

**Access modifier has impact on method overriding.**

**Access modifier does not have any impact on method over loading.**

**-------------------------------------------------------------------------------------------------------------------**

The access level cannot be more restrictive than the overridden method access level. For example: if the super class method is declared public then the overriding method in the sub class cannot be either private, default or protected.

-------------------------------------------------------------------------------------------------------------------------

when overriding a method from a superclass in a subclass, the access level (visibility) of the overriding method cannot be more restrictive than that of the overridden method in the superclass. This principle is known as the "access level rule" or "visibility rule" for method overriding.

-----------------------------------------------------------------------------------------------------------------------------

**Here's a breakdown of how the access level rule works:**

1. If the superclass method is declared as **public**, the overriding method in the subclass can be **public or protected,** but not **default or private**.
2. If the superclass method is **declared as protected**, the overriding method in the subclass can be **public or protected**, but not default or private.
3. If the superclass method is declared as **default** (package-private), the overriding method in the subclass must also be **default or protected**, but not public or private.
4. **If the superclass method is declared as private, it cannot be overridden in the subclass because private methods are not accessible outside the class in which they are declared.**

* **If a method cannot be inherited then it cannot be overridden, therefore, private methods/fields cannot be overridden.**
* **A method declared static cannot be overridden .**

Because static method belongs to class, the don’t change their values even in subclasses.

* **Main method cannot be overridden because main method is static.**
* **A method declared final cannot be overridden because encapsulation.**

**In overloading we have to check only methods names (must be same) and arguments types (must be different) except these the remaining like return type access modifiers etc. are not required to check**

**But in overriding everything matters like method names arguments types return types access modifiers etc.**

**--------------------------------------------------------------------------------------------------------------------**

**What is static/compile time/binding polymorphism???**

**Compile time polymorphism in which compiler decide which method to call.**

In overloading of methods, the compiler decides which overloaded method to call based on the number, types, and order of parameters. this decision is made at compile time it's called static polymorphism.

Compile time refers to the period during which a program is being translated from human-readable source code into machine-executable instructions by a compiler.

**------------------------------------------------------------------------------------------**

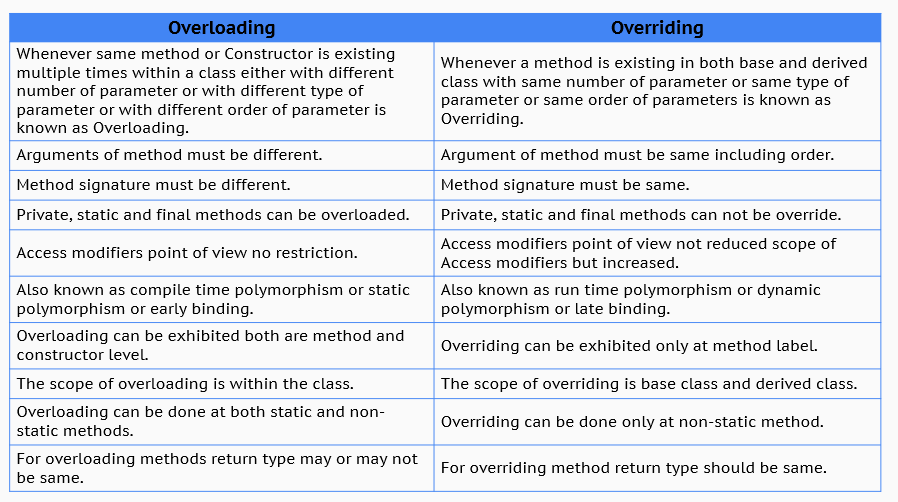
**What is dynamic/run time/ polymorphism????**

**Run time polymorphism in which jvm decide which method to call from parent class or from child class.**

in overriding of method the Java Virtual Machine (JVM) decided which method to be called from parent class or from child class. This process of choosing the right method during program run time is called dynamic polymorphism or runtime polymorphism.

Runtime, also known as execution time, refers to the period during which a program is executed by a computer's processor.

**---------------------------------------------------------------------------------------------------------**

****

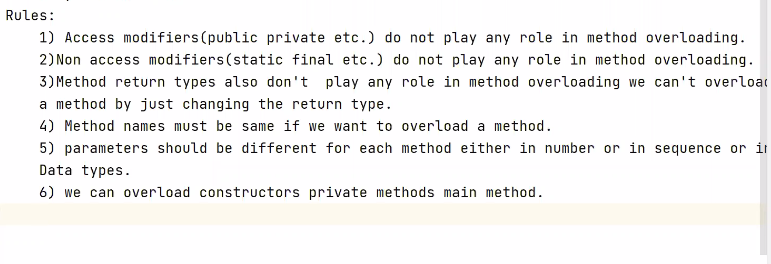
**Static**

**method does not participate in inheritance because each time object creation occur. each object has its own values of variables.**

**Read slide number 268 very important.**

**=====================================================================================================================================================================================================================================================================================================================================================================================================**

**JAVA CLASS 21 TOPIC POLYMORPHISM**



**Where polymorphism is used??????**

**When you have multiple same methods in multiple class and you want to execute all of them then use polymorphism.**

Polymorphism is a fundamental concept in object-oriented programming (OOP), and it is used in various contexts to enhance code flexibility, reusability, and maintainability. It is used for method overloading method overriding.

The things which do not participate in inheritance (constructor, static, final, private) can participate in method overriding.

-------------------------------------------------------------------------------

**What is polymorphism?**

The process in which same methods acts/behave differently on the basis of objects (\*) ,is called polymorphism.

**basis of object mean if overloading while creating object of class , you will pass values by creating object, if you will pass int first method will call, if you pass double second method will call, similarly for method overriding , if you create object of child class then method from child class will be called , if you create parent class method from parent class will be called.**

------------------------------------------------------------------------------------------------------

**There are two types of polymorphism**

1. Polymorphism which is done by method overloading is called **compile time/static/early binding polymorphism**

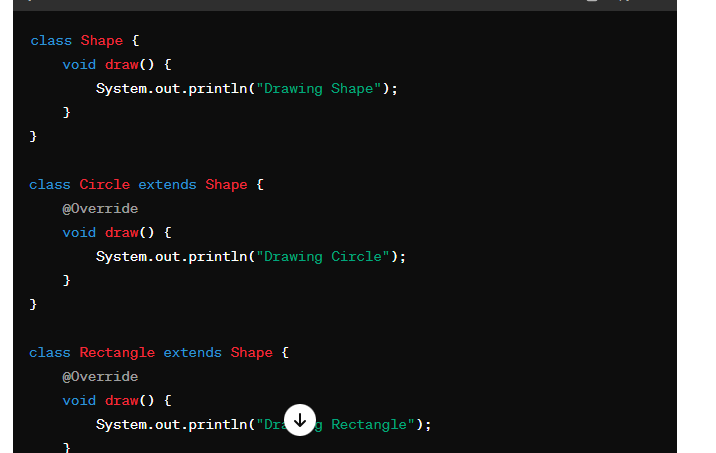
Complier decide which method has to execute at compile time based on the method signature.

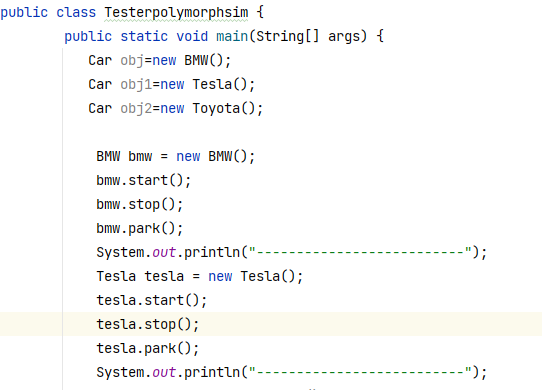


----------------------------------------------------------------------------------------------------------------------------

2. Polymorphism which is done by method overriding is called **dynamic/run time/late binding polymorphism**

JVM decided which method has to be called base on object of parent class or child class this is dynamic polymorphism.





==============================================================================================================================================================================================================================================

**JAVA CLASS 22 FINAL KEYWORD DATE; 14-03-2024**

**What is final keyword???????**

**Final keyword is used to make the things unchangeable.**

**Final keyword is used with variables, methods and classes.**

**Final variables -------- to keep the variable constant**

**Final methods -------- to prevent method from overriding**

**Final class ------------- to prevent the class from inheritance**

=============================================================================================================================================================================================================================================================================================================================================================================================================

**JAVA CLASS 23 ABSTRACTION 14-03-2024**

**Real example of abstraction is atm card.**

**Abstract mean incomplete.**

**abstraction is a fundamental concept of Object-oriented programming (OOP) that allows developers to mode real-world entities as classes and objects in their code. Abstraction involves focusing on the essential qualities of an object while hiding the irrelevant details.**

**What is abstraction????**

Abstraction is a process **of hiding the implementation details and showing only functionality to the user**.

Abstract methods show functionality and their implementation is done in subclasses.

Without abstract keyword other methods are directly implemented in the class.

-----------------------------

For example, sending sms, you just type the text and send the message. You don't know the internal processing about the message delivery.

Abstraction is the process of separating ideas from their action.

**What is abstract class????**

**A abstract class is one which is declared with abstract keyword and can contain defined methods and undefined method**

abstract class is an idea , template plan from which subclasses can be created.

----------------------------------------------------------------------

**What is concrete class????????**

A concrete class is a class that can be instantiate directly by create objects.

**Interview question???**

Abstract’s class object can-not be directly created.

Because it will contain undefined methods, and object of undefined methods’s class will be incomplete.

---------------------------------------------------------------------------------------------------

Is it necessary an abstract class has abstract method?? no

Can an abstract method must be declared inside the abstract class, yes.

Abstract not allowed with variables.

Abstract no allowed with constructor, because not inheritance.

Static variables can be accessed inside the constructor.

**How to achieve Abstraction?**

**1. by abstract classes**

**2. by interface**

**Important points of abstract class**

We use abstract classes when we know the methods, but we don’t know how they would be implemented.

Every abstract class participate in inheritance.

Abstract class of java always contains common features.

Abstract classes definitions should not be made as final because abstract classes always participate in inheritance classes.

An object of abstract class can not be created directly but it can be created indirectly.

=============================================================================================================================================================================================================================================================================================================================================================================================================

**JAVA CLASS 23 INTERFACE DATE;15-03-2024**

**What is partial abstraction????**

An abstract class has both abstract methods and concrete methods, but abstract class forces to subclasses provide implementation for abstract methods, and do not provide any force for concrete methods, to override them or not, this is lack against rule. That’s why this is called partial abstraction.

**What is fully abstraction???**

The class which only have abstract methods this happens in interfaces.

But from java version 8, interface support default and static methods. That’s why this is not more fully abstraction. But you will consider this fully abstraction.

**What is difference in abstract class and interfaces???**

**Abstract Class**:

* + Can contain both abstract and concrete methods.
  + Can have instance fields.
  + Can extend only one abstract or concrete class.
  + Cannot be instantiated directly; needs to be extended by a subclass.
  + Provides a partial implementation of a class, with the intention of being extended by subclasses to complete the implementation.
* **Interface**:
  + Can only contain abstract methods, constants, default methods, and static methods.
  + Cannot have instance fields.
  + Can extend multiple interfaces.
  + Cannot be instantiated directly; needs to be implemented by a class.
  + Provides a contract for classes to implement, defining a set of methods that implementing classes must provide.

-----------------------------------------------------------------------------------------------------------

You can not create the object of interface and abstract directly, because they have incomplete methods.

Using final variable inside the interface.

**What is an interface???????**

Interface is similar to class which has public static final variables (constants) and public abstract methods only. It is used to achieve fully abstraction and multiple inheritance in Java.

You can implements multiple interfaces, resolve diamond problem. Multiple parent one child.

The interface is also a mechanism to achieve abstraction in java.

There can be only abstract methods in the interface (before java 8) Now we can have default and static methods as well.

--------------------------------------------------------------------------------------------------------------

**Rules for interfaces????**

1. By default Interface is abstract we don’t need to write abstract keyword
2. By default variables in interface are public static final (constants) you don’t need to write public static final keywords.
3. By default methods in interface are public abstract methods you don’t need to write public and abstract keywords.

---------------------------------------------------------------------------------------------------

**What is the difference in class and interface?????????**

**An interface is not a class. Writing an interface is similar to writing a class, but they are two different concepts.**

**In a class attributes and behavior are implemented by creating object of a class.**

**In interfaces attributes and behavior are implemented by creating a subclass class.**

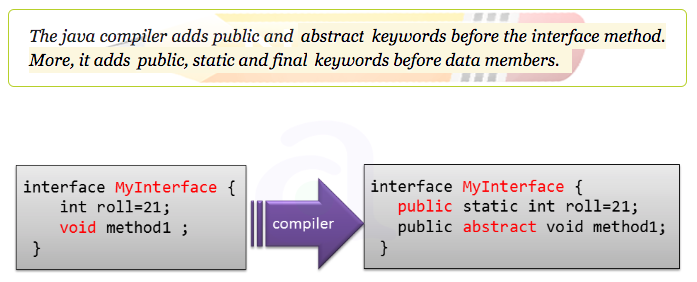
**-------------------------------------------------------------------------------------------**

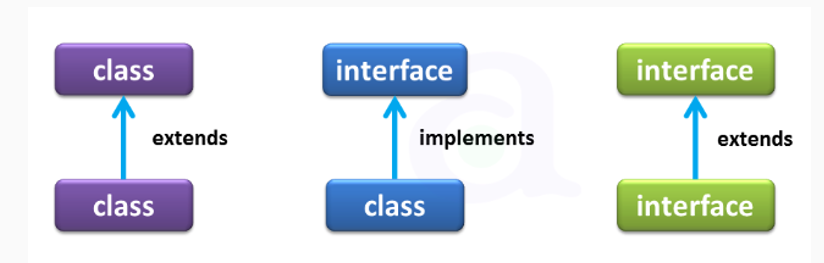
Interface can extend multiple interfaces. It means interface support multiple inheritance

**When use abstract class, when use interface???**

When you want complete abstraction, you don’t know about methods implementation use interface.

When you partially know about implementation use abstract class, some defined and few undefined method.







=========================================================================================================================================================================================================================================================================================================================================================================================================================================================================================

**JAVA CLASS 24 ENCAPSULATION 15-03-2024**

Encapsulation is one of the four fundamental OOP concepts.

**What is encapsulation??????????**

Encapsulation is a mechanism/process of wrapping the variables and methods together as a single unit.

typically a class. This unit acts as a capsule or container.

Encapsulation is achieved in java language by class concept.

Combining of state and behavior in a single container is known as encapsulation.

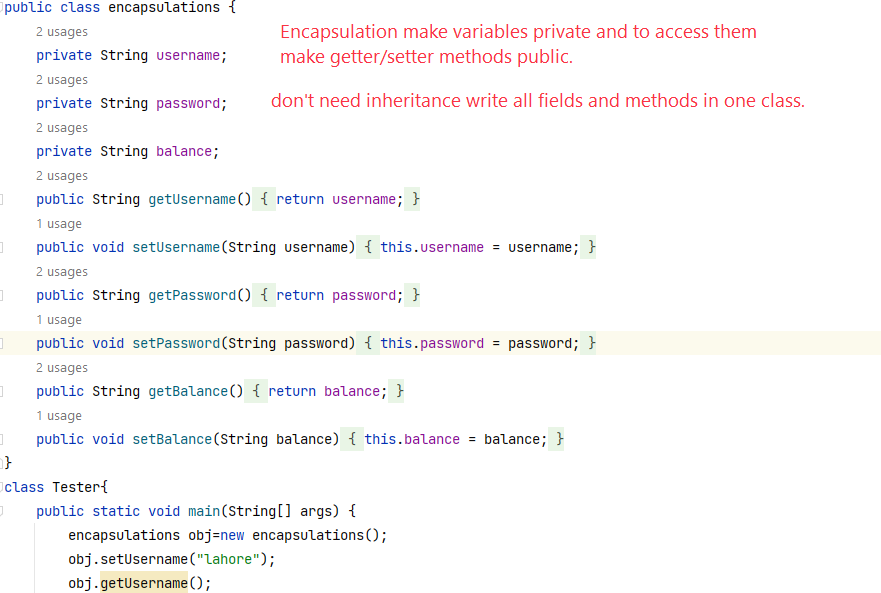
In encapsulation, the variables of a class will be hidden from other classes and can be accessed only through the methods of their current class. Therefore, it is also known as data hiding.

**What are getter and setter methods?????????**

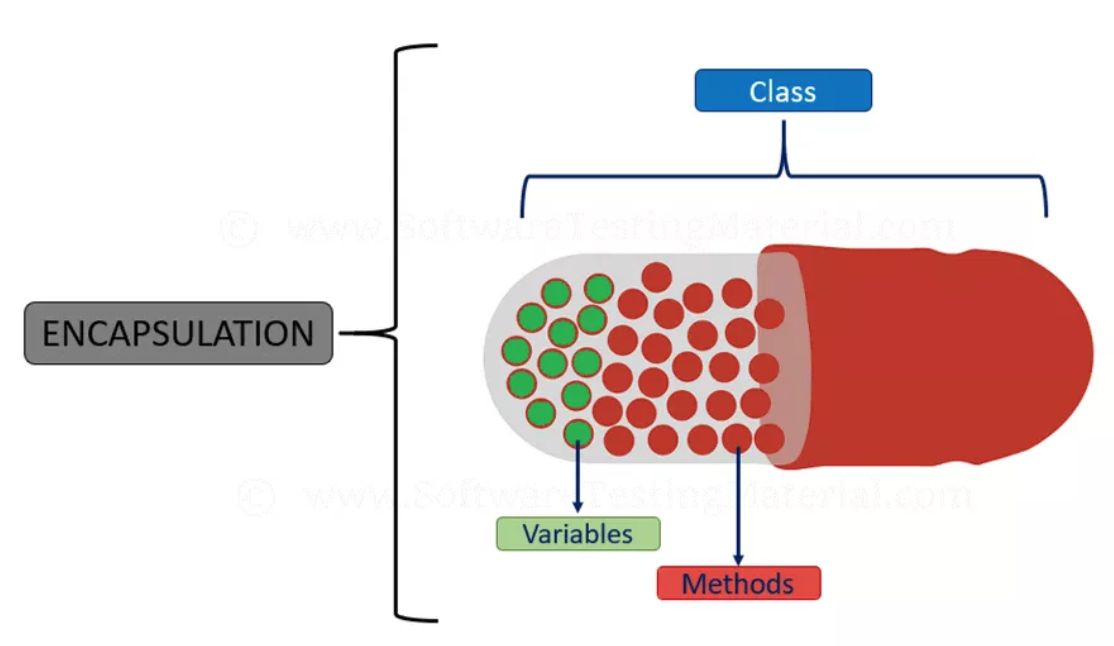
Getter methods are used to get or access the values of the private variables of an object.

Setter methods are used to set or update the values of the private variables of an object.

------------------------------------------------------------------------------------------------------------------------------------===================================================================================================================================



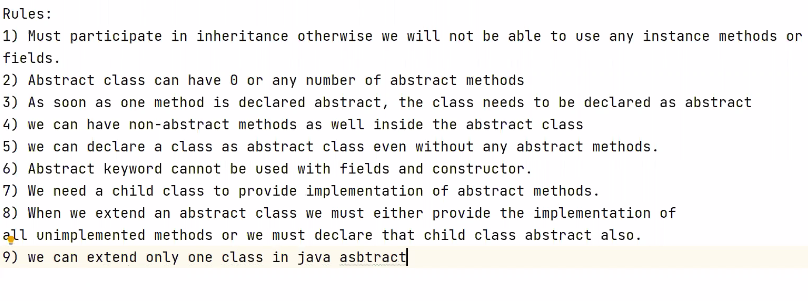
Encapsulation variables are set private, and methods getter, setter are set public. You don’t need inheritance for encapsulation.

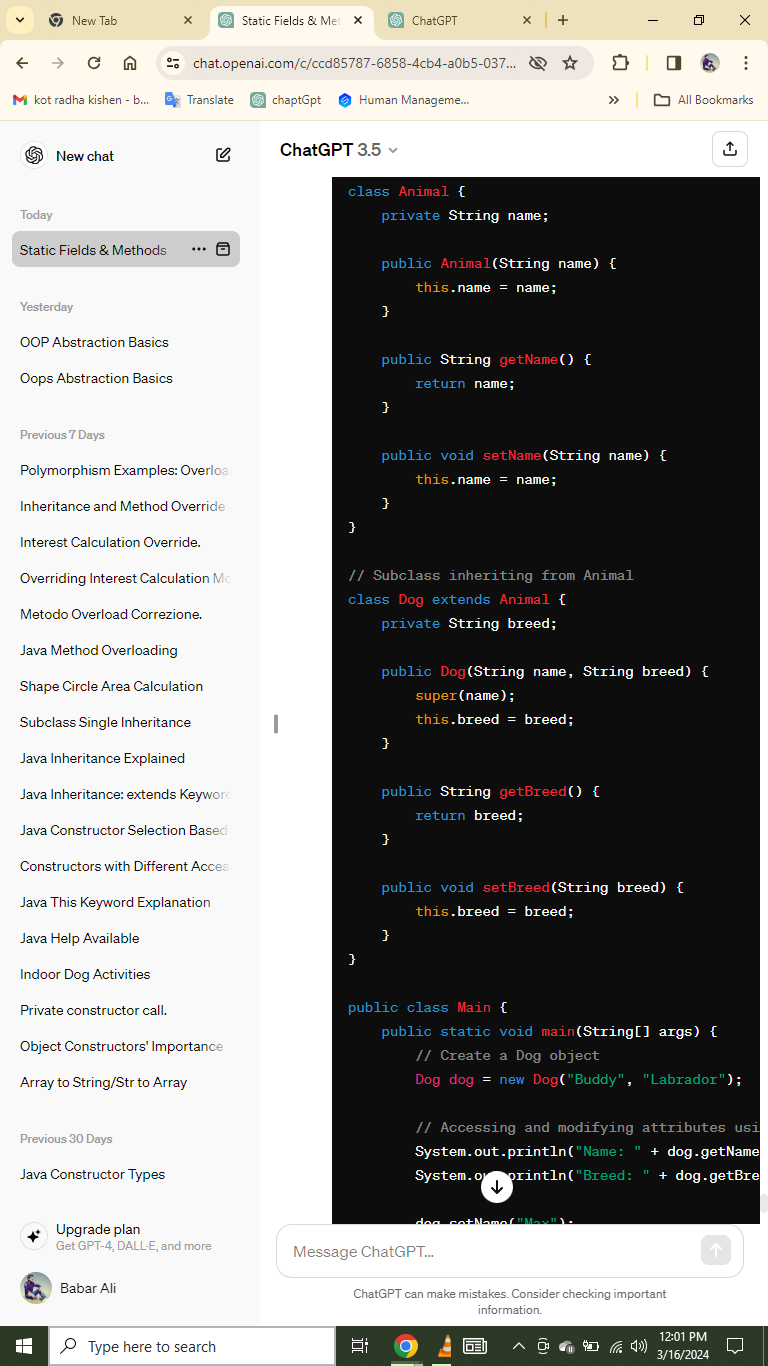


To achieve encapsulation in Java

* Declare the variables of a class as private.
* Provide public setter and getter methods to modify and view the variables values.

==============================================================================





=====================================================================================================================================================================================================================================================================================================================

Very importance

Inheritance

In inheritance suppose there are dog class, cat class, cow class, these all are animal and these all have some common properties, instead of writing separate classes we make a animal class which has all the common properties of all animals. In this case you can add more properties in common properties also. Save code duplication, save memory , save time . you can access all the properties of animal class and can add more functionalities. But you can not make changes in common properties which are present in animal class. This you can do in method overriding feature of java.

Overriding. Suppose you inherit the superclass and you now have access of all members of super class but some of methods you don’t like of super class and want to make changes in them , this is done by overriding.

Abstraction is separate concept. Suppose you make some things visible and some things hide for user . then you use abstract class , in which some things are hidden and some methods are visible to all .

Interface. When you want to hide all the things from user, then you use interface. Basically interface is the contract with subclasses that if you want to implement the super class you must implement the methods in super class.

Encapsulation. This is another concept. This is used to hide the data member so that no one can access from outside.

================================================================================================================================================================================================================================================================================================================================

**CLASS JAVA 25, WRAPPER CLASSES date; 16-03-2024**

**If I have to store a value , we store this inside a variable. If I have to store lot of variables I should go for arrays. Because each time write name of variable and storing value , a lot of redundant code. So we use arrays.**

**But there is a some limitation for arrays also.**

**They are fix in size.**

**--------------------------------------------------------------------------------------------------**

**Arraylist are classes, we use add() method to put method inside array class.**

**--------------------------------------------------------**

* **Fixed Size**: Arrays can't change their size after creation, which can be limiting if you need to adjust the size during the program. If you fix size 100 people and people come 80 on a wedding then 20 seats will be reserved in memory will be waste.
* **Store One Data Type only**: Arrays can only hold one type of data, so you can't mix different types in the same array.
* **Index Out of Bounds**: If you try to access a value in the array that doesn't exist, you'll get an error. Searching of an element , remove element , insert element there is no method in java. . this is problem it is difficult because you don’t know the index of an element at which position this present whom you want to change, update, or delete.

--------------------------------------------------------------------------------------------------------------------

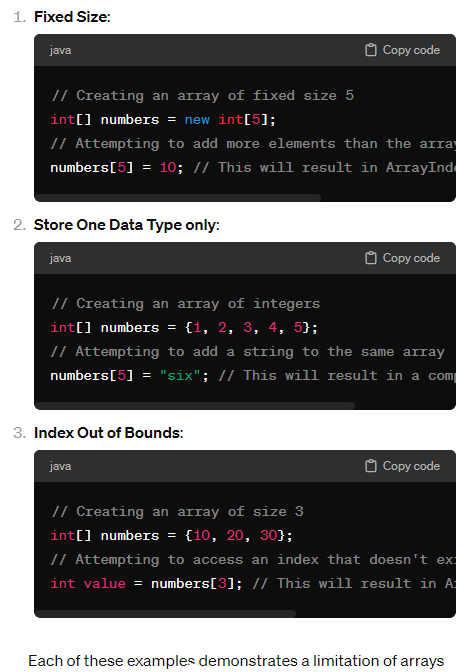
Arrays are useless??? No arrays are very fast, use very less memory, use always arrays when you know the exact size then use arrays.

=============================================================================

**Why we need wrapper classes????**

**Because collection frame work accepts only objects of any classes, But primitive data types are not classes they are just data types. So to convert the primitive data types into object there are wrapper classes, wrapper classes convert the primitive into objects so that you can use them inside collection frame work.**

**if you have an int primitive and you want to add it to a List, which requires objects, you can use the Integer wrapper class to convert the int primitive into an object:**



**What are wrapper classes????**

**Wrapper classes are used to convert primitive data types into corresponding objects of wrapper classes.**

**What is auto-boxing???**

Autoboxing is the automatic conversion of primitive data types into their corresponding wrapper class objects.

Integer num=20; >>>>>>> this is auto boxing

**What is auto-unboxing???????**

Converting objects of wrapper classes into primitive data type is called auto un boxing.

Auto-unboxing is the automatic conversion of wrapper class objects back into their corresponding primitive data types.

Integer num=20; this is auto boxing

Int value=num; this is auto un boxing.

===========================================================================================================================================================

**What is a collection?????**

 A collection is an interface that represents a group of objects. List, Set, Map are group of objects.

-------------------------------------------------------------------------------------------------------------------

Similarly an array list is an object of books, and you can perform task like, add more book, remove book, update book inside the collection.

----------------------------------------------------------------------------------------------------------------------

**What is collection frame work??**

A framework is a set of classes and interfaces which provide a ready-made architecture.

----------------------------------------------------------------------------------------------------------------

**Why we use collection frame work??? For data manipulation.**

Java Collection Framework enables the user to perform various data manipulation operations like storing data, searching, sorting, insertion, deletion, and updating of data on the group of elements

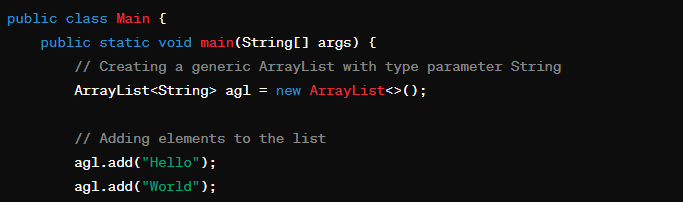
**Interface are contract with classes,**

**Interfaces are list, set, map**

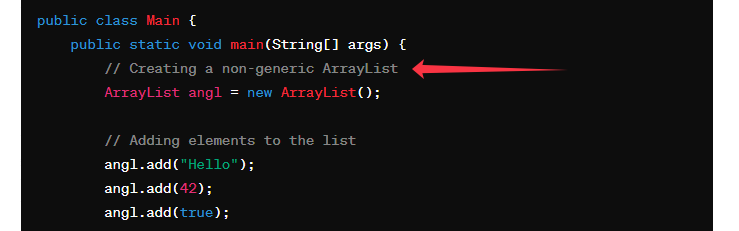
**And their implementation is done by classes, array list, linked list, hash set, hash map.**

Generic collection , you can store only one type of data.

Java new generic collection allows you to have only one type of object in the collection.



Non generic , before jdk 1.5 , you can store multiple type of dat , lik int, string, double.



===============================================================================================================================================================

Array list add method; number.add(index 3, add value 49) this will add new element at index 3.

There is no method in array.

Combining two array list together. Use add all (number); method.

================================================================================================================================================================================================================================================================================================================================

**JAVA CLASS NUMBER 26 , ARRAYLIST**

**What is a list???**

**List is an interface which is an ordered collection of elements that can contain duplicate elements**

A List is an interface in Java that represents an ordered collection of elements.

**List is one of the most used Collection type.**

**Classes that implement List interface:**

* **Array List**
* **LinkedList**
* **Vector**

**-------------------------------------------------------------------------------------------------------------------**

**What is an array list???**

**Array List is a class which implements the List interface of collection framework.**

**Array List can contain the duplicate elements and order of elements is maintained.**

**Implementation**

**Duplication**

**Insertion order**

RetainAllmethod() in two list , which will be common in both will be print.

word.retainAll(sentence); this mean that there are two array list one is word and second is sentence , retain all mean that , keep all elements of word array which are present in sentence , and remove all other.

-----------------------------------------------------------------------------------------------------------------------------

**Array only were fixed in size.**

**Array store only elements of one data type**

**Array List is dynamic in size.** meaning it dynamically adjusts its size as elements are added or removed.

Array list can store multiple type of elements.

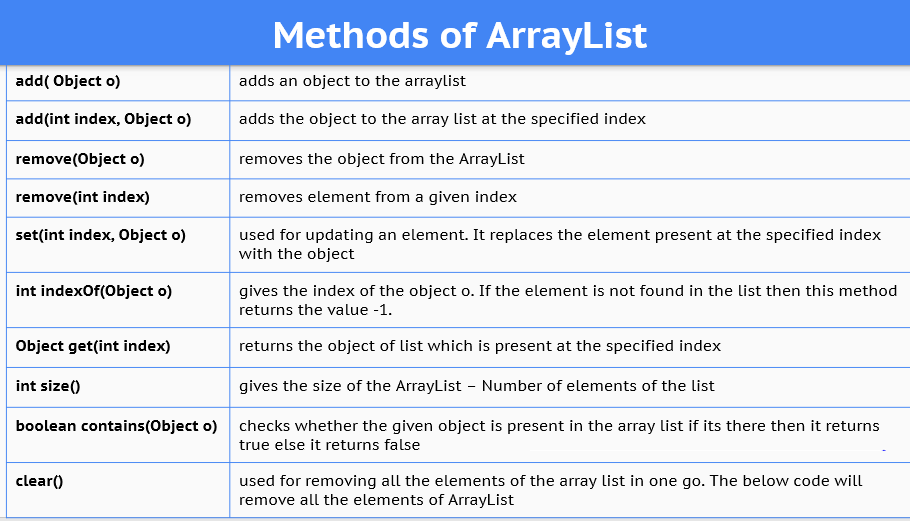
---------------------------------------------------------------------------------------------------

**Very important question??**

**Don’t use loops when your data changing occur of array list. Otherwise you can use.**

With array list you should not use loops , use iterator. Because if you are doing two parallel things with loops like accessing an element and same time removing element then loops will give error will create bugs.

Because loops don’t work with dynamic data. They work with static data which is not changing.



================================================================================

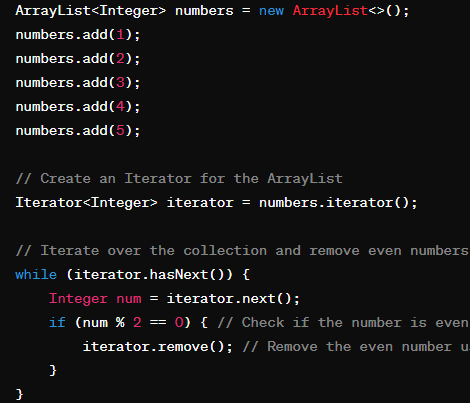
**What is iterator???????**

Iterator is an interface which is used to iterate the elements in collection frame work.

Why can not use loops, when the size of array list is changing ,loops work with static data perfectly. Iterator work with dynamic data only.

**Iterator has 3 methods.**

1. **Has next method.** it gives true if there are element inside the array list
2. **next** method move the pointer from index zero to next index. Initially pointer is at -1 index.
3. **Remove method** is used to remove element from array list.



**What is difference in loop and iterator????**

**Iterator use has next and next method to iterate ,**

**Loops used, size method, Boolean and condition to iterate elements.**

-----------------------------------------------------------------------------------------------------------

**What is linked list???????**

**linked List is a class which implements the List interface of collection framework.**

**linked List can contain the duplicate elements and order of elements is maintained**

**Implementation**

**Duplication**

**Insertion order**

--------------------------------------------------------------------------------

**What is difference in array list and array????**

Arrays are fixed in size

Arrays can store only one type of elements.

Arraylist are dynamic in size

Arrays use = sign to store data arralist use add method to store data

Arrays use length method arraylist use size method to check the size

----------------------------------------------------------------------------------------

**What is difference in array list and linked list???**

The only difference is that how they store the data internally.

**How array list and linked list store data??**

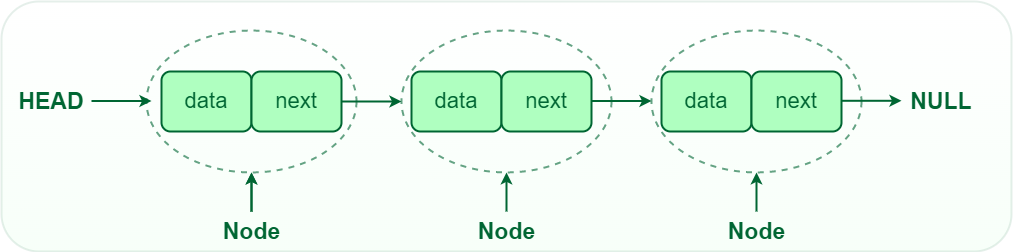
LinkedList internally uses a **doubly linked list data structure** to store elements.

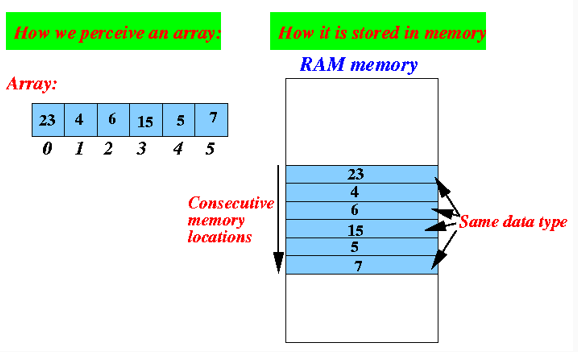
A doubly linked list is a type of linked list where each node contains two pointers: one pointing to the previous node and another pointing to the next node.

Array List internally uses an array to store elements **in consecutive memory locations**.

**Array list store the data in the form of consecutive memory location.**

**Linked list store the data int the form of doubly linked list.**



****

**When should use array list when should use linked list??**

**When you want to only access the data use array list.**

**When there is lot insertion and deletion of elements use linked list.**

--------------------------------------------------------------------------

. **When there is lot of insertion and deletion of data use linked list and use iterator.**

**When there is only you want to access data use array list and use advance loop**

-----------------------------------------------------------------------------------------------------------------

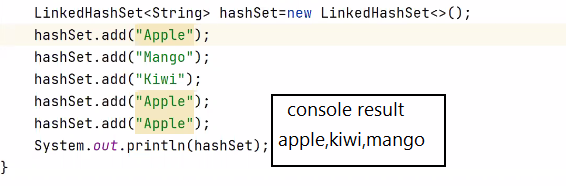
=============================================================================================================================================================================================================================================

**What is set???**

Set is an interface in collection framework that can’t contain duplicate elements and don not maintain the order.

**Classes that implement Set interface**:

1. **HashSet**  --------------------------is a class that implements set interface. don’t allow duplication, don’t maintain Insertion order.
2. **Linked HashSet**-------------------- is a class that implements set interface. don’t allow duplication , and allow to maintain insertion order.
3. **Tree Set**------------------- is a class that implements set interface. don’t allow duplication, and Sorting of elements alphabetically or numerically in case of integer ascending order.



**Implementation**

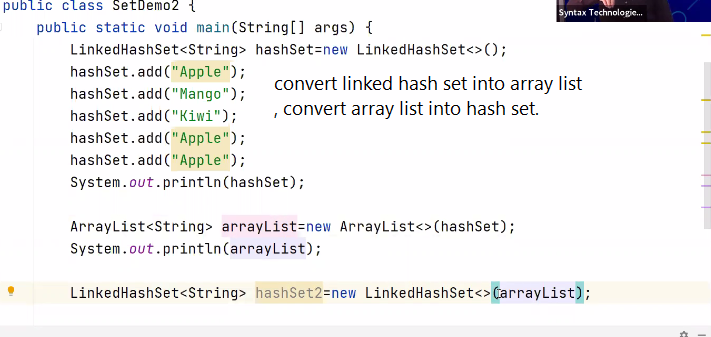
**Duplication**

**Insertion order**

**HashSet**, class that implements Set interface. It does not allow duplicates and does not guarantee any insertion orders. It allows null elements.

**Linked HashSet**, class that implements Set interface. It does not allow duplicates and orders its elements based on the order in which they were inserted.

**Tree Set**, is similar to HashSet except that it sorts the elements in the ascending order while HashSet doesn’t maintain any order.



===============================================================================================================================================================

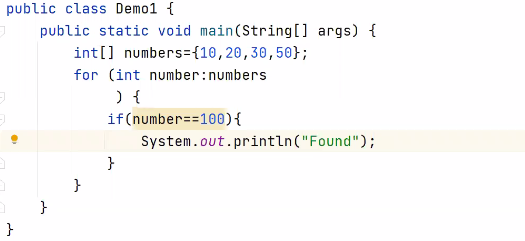
============================================================================================================================================================================================================================================

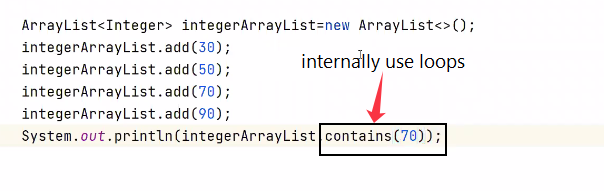
JAVA CLASS 28 TOPIC MAPS DATE-RAMZAN= 18-03-2024

Scenario ; in arrays if you want find any element by using loop, it will first search an compare the values which you want element. If a list contain 3 million elements it will take lot of lot of time.

Similarly in array list, if you want to search any element using **contains method** it will first search and then will compare and then will give, internally contain method is also using loops. It will take a lot of lot of time.

To solve these both problems there is map interface in java.





-----------------------------------------------------------------------====================================================================

**In map we create own indexes**

**In array list and array computer create itself indexes.**

==============================================================================================================================================================

**What is a map???????**

**Map is an interface, which store the elements in the form of key-value pairs.**

Classes which implements the map interface.

1. **Hash map** --------------------------is a class that implements map interface. : HashMap does not allow duplicate keys, but it allow duplicate values.
2. Insertion Order: HashMap does not maintain the insertion order of key-value pairs.
3. **Linked Hash map**-------------------- is a class that implements map interface. Linked Hash Map does not allow duplicate keys but allows duplicate values. Linked Hash Map maintain the insertion order of key-value pairs.
4. **Tree map**------------------- is a class that implements set interface. Tree Map does not allow duplicate keys, but it does allow duplicate values.
5. Tree map Sorting of elements alphabetically or ascending order.

**Implementation**

**Duplication**

**Insertion order**

===============================================================================================================================================================

**addAll()** method is used to add all elements of a collection to another collection. This method is commonly used with ArrayLists, but it can be used with any class that implements the Collection interface.

the **removeAll()** method is used to remove all elements from a collection that are also present in another collection. It's commonly used with lists like ArrayList or LinkedList, but it can be used with any class that implements the Collection interface.

-----------------------------------------------------------------------------------------------------------------------

===============================================================================================================================================================

1. **Put method**:

Adds a key-value pair to the map or replaces the value if the key already exists.

1. **Get method**:

Retrieves the value associated with the specified key from the map.

1. **Replace method**:

Replaces the value associated with a specified key in the map.

1. **Is empty method**:

Checks if the map is empty, i.e., it contains no key-value pairs.

1. **Contains method**:

Checks if the map contains a specific key-value pair.

1. **Contains value method**:

Checks if the map contains a specific value.

1. **Contains key method**:

Checks if the map contains a specific key.

Putt all method ----- putting one map into another map

------------=============================================================================================================================================================

**Iterators in map ???**

**There are three types of iterator in maps**

* **Iterator to get the values only**
* **Iterator to get the keys only**
* **Iterator to get the both key-value pairs**

**Keyset-**-----is a method used with set interface which gives all the keys of a map.

Iterator<String> keys=fruit Map . key Set().iterator;

System.*out*.println(keys);

--------------------------------------------------------------------

**Values()**--------is method with collection interface to get all the values of map

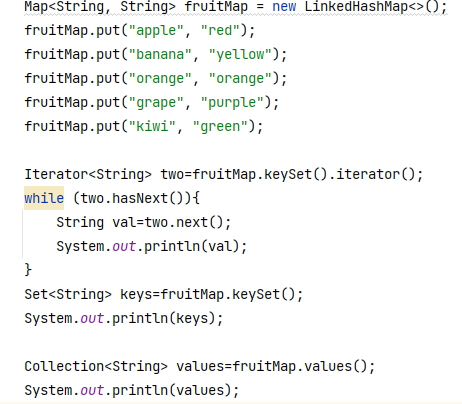
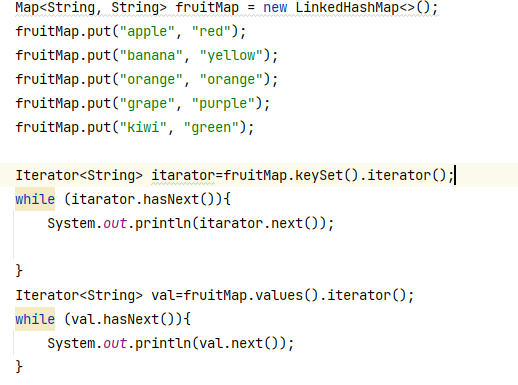
Collection<String> values=fruitMap.values();  
System.*out*.println(values);

**Entry Set-**---------- there is used with iterator of map entry method. and use for lop and get values and then get key methods.

Iterator<Map.Entry<String,String>> all=fruitMap.entrySet();  
while (all.hasnext){  
Map.Entry<String,String> val=all.next();

String sum=Val.getKeys();  
}

-===========================================================================





========================================================================================================================================================================================================================================

===============================================================================================================================================================================================================================================================================================================================

**JAVA CLASS 29, CONFIG FILES DATE;18-03-2024**

**To read the data from a file**

* Path of the file
* Software which will read or write the data from that file, ms office, adobe , image viewer,
* Knowledge of structure of file.

**Structure of excel file**

* **Workbook**
* **Sheet**
* **Rows**
* **Columns**

**-----------------------------------------------------------------------------------------------------------------------------------**

**What is file input stream and file output stream????**

**File Input Stream and File Out put Stream classes comes from java.io library.**

**fileInputStream.close();**

**File Input Stream is used to read data from a file.**

**Java File Out put Stream is used for writing data to a file.**

**--------------------------------------------------------------------------------------------------------------------------**

**Why we do need file handling???**

**1. for security reason**

**2. for accessing the data of out side files.**

**Why we write data inside files, why do we need files???**

**1.** We write data inside files for security purpose so that no one can read the data from file , once you test your data then you delete the files.

**2.** We need files handling so that we can access data from out-side the source code.

When testing web forms, all testing data is not put inside the code. Instead, we create files to read data from excel file.

-------------------------------------------------------------------------------------------------------------------

How create property file inside intellij.

1. right click on project>select directory>click on directory>new file>config.properties

----------------------------------------------------------------------------------------------

**What is properties class in java???**

This class is used to read the files of having extension (. Properties)

**Properties class have similar method of like map**

**1. put method-----put method is used while write data**

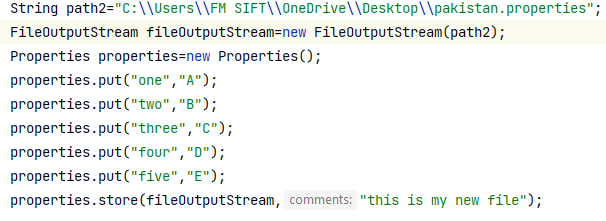
**2. get method-----is used when get data read data from file**

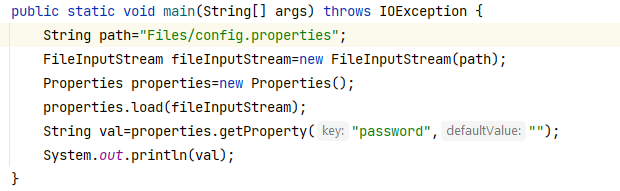
**3. load method --- to read data ---load—file input stream variable**

**4.store method----to write data---store ---file output stream variable**

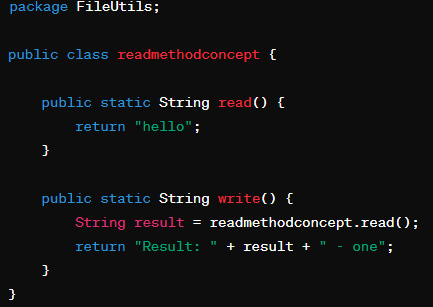
**What is a property file????**

The file which store the data in the form of key and values



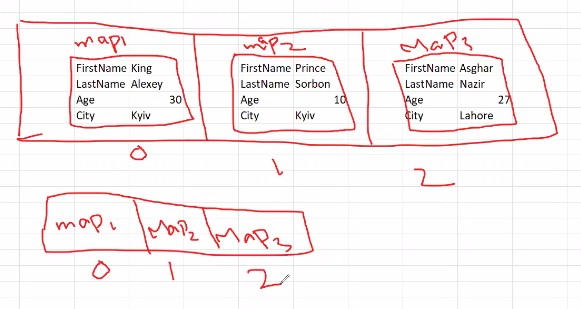


Calling a method inside other method, calling a static return method inside another static return method. method calling.



================================================================================================================================================================================================================================================

**JAVA CLASS 30 , EXCEL FILE READING. Date;20-03-20424**



========================================================================================================================================================================================================================================================================================================================

**JAVA CLASS 31 EXCEPTION HANDLING 20-03-2024**

**What is exception????**

An **Exception** is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's

**What is exception????**

When something unexpected has happened during the execution of a program.

The process of converting system error messages into user friendly error message and providing alternate solution is known as Exception handling.

===============================================================================================================================================================

**At a time only one Exception is occurred and at a time only one catch block is executed.**

**Common exceptions**

1. Arithmetic Exception
2. Null Pointer Exception
3. Array Index Out Of Bounds Exception
4. File Not Found Exception
5. IO Exception
6. Interrupted exception --------thread. sleep

**What is try and catch block?????**

Try and catch block is similar if else condition, which mean try any statement if there is an error then please give it to catch block, the catch block catches the error and then print that error, do not terminate the program