c is procedure-oriented, garbage collection in c is manually

java is object-oriented,

Anything Java depends on objects, data abstraction,

polymorphism

It's saving and faster in execution

class loader is part of the java runtime environment responsible for java class into the java virtual machine

it helps developers to create compile packages and run Java applications it proved essential for Java developer

responsible for running Java applications executing Java application

in JVM-interpreted Java this process allows Java applications to be platform-independent

concrete implementation refers to the specific fully defined implementation of the interface

runtime instance refers to an object that is created and exists in memory

it's the physical component of a computer system that directly executes tasks without the abstract layer

when a program language supports dynamic typing it means that type checking performs at run time

public accessible from any other class

private accessible within the class which is declared

protected accessible within the same class and sub-class and in package

contract name same as the class name. contructor don't have return type.

A constructor can be overloaded but can’t be overridden.

Method overloading in Java is a feature it is always a class that has more than one method with the same name but with different parameters java support method overloading throws two mechanisms to make changes in the parameter it can be with different no of parameters different types

Method overriding in Java occurs when a subclass as the same method as the parent class in other words method overriding occurs when a subclass provides a particular implementation of a method declared by one of its parent classes

A destructor is used to destroy the object

This keyword is used to the current value or current state. This keyword also refers which refer the object's local variable name to the instance variable name.

This is a keyword in Java, that point to the current object,

Member functions in Java also known as are blocks of code within a class that performs a specific class known as a member function.

Shadowing occurs when a local variable or a parameter has the same name as an instance variable

Accessor geters:- to retrieve the value of a private instance variable

vs

mutator seter:- to modify the value of a private instance variable

improve readability and reduce the complexity

Inheritance

Single-level, multiple, multi-level, hierarchical, hybrid these types of inheritance

Encapsulation is a data hiding mechanism, grouping together data and methods acting according to data

Encapsulation in Java is a process of wrapping code and data together into a single entity ex. Capsule.

By providing only the getter and setter method you can make the class read-only or write-only

It is way to achieve data hiding in Java because other classes will not be able to access data from other classes

The encapsulation class is easy to test so is better for unit testing

Polymorphism in Java is a concept by which we can perform a single action differently.
Polymorphism driven from poly means many and Morphis means form

There are two types of run time and compile time

Runtime poly:-not determined by the compiler, obtained by virtual function & pointer

Compile time:- in this call is determined by the compiler, it is obtained by operator overloading and function overloading

This runtime poly also known as early binding and static binding

This compile poly also known as let and dynamic, overriding binding

Abstraction is the process of hiding the implementation details and showing only functionality to the user another way is to show only essential things and hide things eg. sending SMS but don't knowing the process

Way to achieve abstraction

abstract class, we can achieve 0 to 100% abstraction

interface 100% abstraction

a class that is declared as abstract is known as an abstract class.

It is an abstract and non-abstract method

The object is an entity which has a well defined by attributes or properties

In a car object, the state includes colour, size

Behaviour refers to in-car object behaviour including start and stop

The identity of an object differentiates it from all other objects even if they have the same state

Responsibility is often related to object behaviour and can be doc as part of the design process

Static variable when a variable is declared static in Java programming it means that the variable belongs to the class it self rather than any specific instance of the class

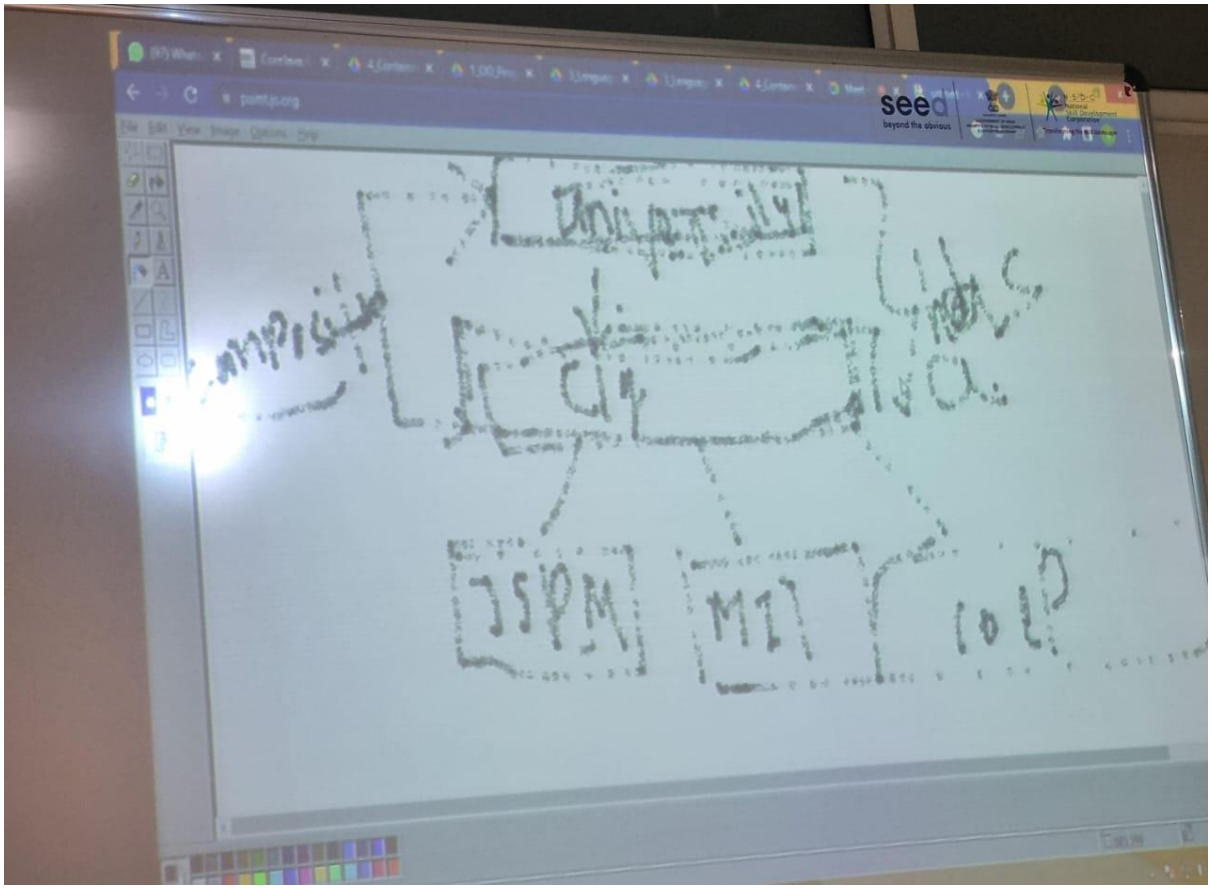
Static method in java is part of java class, every instance of class variable access class

Static block in any block we use static key word then it becomes a static class.

Composition is a way to design or implement the as-relationship

Composition and inheritance both are design technic, inheritance is used to implement is-a relationship as-a relationship is used to code the reusability of the program

University and collage relation like university will close then collage also close that type of relation



Aggregation if a class have entity reference it is know as aggregation. Reperestion has-a relationship

Class Employees{
int id;
String name;
Address address : zip code, city, state, area;

//As-a relationship
}

When use aggregation: code reuse is best achieve by aggregation when there is not is-a relationship

Inheritance only useful only if the is-a relation that maintains throw out the life to me of the object invoke otherwise aggregation is best

Containment means composition or has a relationship if the class as a variable other class types which is has-a type

Super the super keyword in java is a reference variable which ia used to refer immediate parent class operator whenever every you create instance of sub class

Super refrence variable usesjava super kay word, super can immadite partent class

Super can be used to invoke immediate parent class method, can be used to invoke immediate parent class

Overloading	overriding
1. Generally done in same class **2.** Handy for program design as Difference method anme need Not be remembered **3.** Difference for each method Overloaded	**1.** In the Inherited classes 2. Message is same but its implementation need to be specific to the derived class 3.has to be same in derived

A java protected keyword is an access modifier that can be assigned to variable methods constructor and inner class

The protected access modifier is accessible within the package have it can be accessible outside the package through only inheritance

We can't assign protection to outer class and interface

If you make any constructor protect you can't create an instance of the class in outside the method

If you override any method, the overridden method must not be restricted

Compile-time binding
overloading, ex:- main

Run-time binding
overriding, ex:- object creation

Dynamic data type:- dynamic type of variable or an expration is the type of its the value during run time when the program get executed * it my change at the program execuation

Telescope

The final kay wor in java used to restrict user . variable method and class in this final is used

If you make any variable as a final you can't change the variable of the final

Stop value change,
stop method overriding,
stop class inheritance

If you make any method as final you can't override.
If you make class as a final you can't extends it.

In java object class is a parent class of all the java classes every java class is a direct or indirect class of a java class, every java class extends the object class

The sub class internally inherites all the object class hence we can say that object class is the cosmic super class in java

Object obj=new Employee():

Public String toString()
public Boolean equals(Object obj)
public int hashCode()
public final class getClass()

protected void finalize()
protected Object clone()
public final void wait(long timout)
public final void notify()
public final void Allnotify()

Concrete Java offers a potent feature known as a class for object-oriented prog
and object may be created using class as a blueprint since both data and behaviour

A concrete class is one that can be instance produce objects in all of its methods including abstract ones inherited from super class or interfaces

The concrete class can be created by a new keyword to an object called direct instantiation

Interface An interface in Java is a blueprint of a class as a static constant and an abstract method the interface in Java is a mechanism to achieve abstraction there can be only one abstraction method in Java there is a body that is used to achieve abstraction and multiple inheritance in Java

Since Java 8 we can have default and static methods in an interface since Java 9 private methods in private method

It can achieve loose coupling
The interface is essentially a collection of Constants and abstract methods.

"programming by contract". Data members in an interface are always public, static and final

A subclass can only have a single superclass in Java but a class can implement any number of interfaces.

If a class implements an interface then have to Implement all the methods of it.

Class-class extends
interface-interface extends
class-interface implements

Before Java 8 interface only abstract method implementation of this method was provided in a separate class if a new method add-in interface then its implementation code as to be provided in the class implementing the same interface to overcome this issue java introduced the concept of default method which allow to implementation

An interface that doesn't contain methods, fields, or constants is known as a marker interface or tag interface. It delivers runtime information to the JVM and compiler, providing additional details about the object. Examples include Serializable and Cloneable.public interface serializable();

Cloneable is an interface that is used to create the exact copy of an object it exists in java.lang package class must implement the Cloneable interface if we want to create a clone of the class object

Serializable

Syntax for instantiating class
OuterClass.InnerClass innerObject = outerObject.new InnerClass();

Anonymous java anonymous inner class is an inner class without a name and for which only a single object is created

It should be used to override a method of a class or interface

Java anonymous inner 1st class is class 2nd interface

A functional interface is an interface that has only one abstract method. From Java 8 onwards, lambda expressions can be used to represent the implementation of a functional interface.

A functional interface can have any number of default methods and static methods, but it can have only one abstract method. Functional interfaces are also known as SAM interfaces (Single Abstract Method interfaces).

Examples of functional interfaces in Java include Runnable, Action Listener, and Comparable.

A lambda expression is a new and important feature of Java which whos included in java it's proven a clear and consistent way to represent one method interface using an expression

In the case of the lambda expression, we don't need to define it by providing the implementation

The string is a collection of character
string is immutable

Compareto() it compare two string lexicographically
equals() content of the two strings same or not

Thread safe code ensures that shared data is accessible and modified in a way that prevents data corruption and unaccepted behaviour and creases typically by using a synchronization mechanism

Java string buffer class is used to create mutable string objects string buffer class in Java is the same as String class accept it is mutable

Java string buffer class thread-safe multiple threads can't accept the same class so it is safe and will result in order

Methods append insert replay, delete reverse, capacity, length, substring

The append method concate the given argument with the given String

String-Builder in Java represents a mutable sequence of characters since the string class in Java creates an immutable sequence of characters

String-builder class is the same as the string-buffer class except it has not synchronised since JDK 1.5

String-buffer 1.0

String-buffer has less effect than string builder

String-buffer is thread-safe and string-builder is not synchronised, not thread-safe

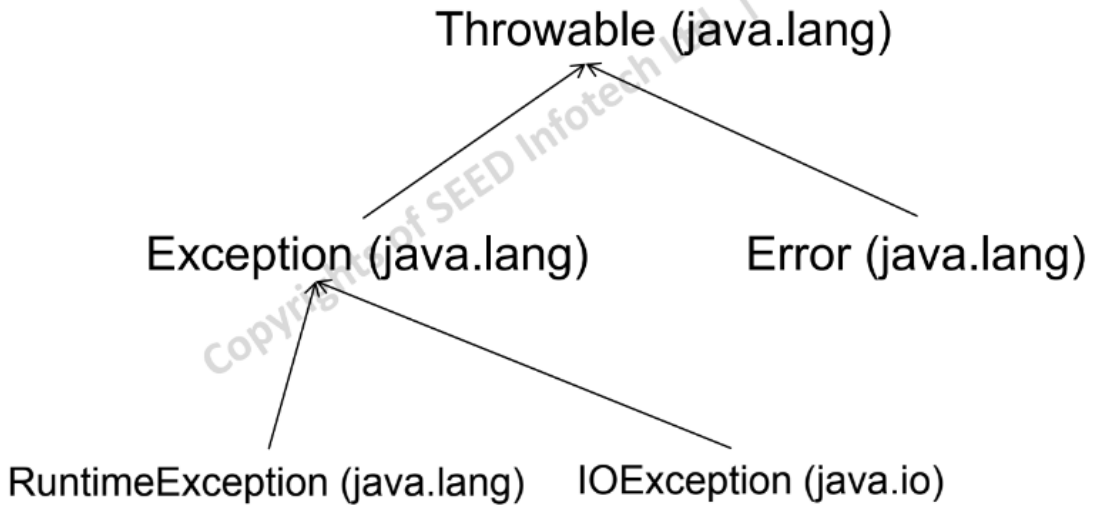
Auto-boxing the automatic conversion of primitive data type into its equivalent wrapper type is known as boxer

Auto-unboxing the automatic conversion equivalent wrapper type into of primitive data type

Exception Handling

An error defines a reasonable issue that stops the execution of the program.

- In java, exception is represented by an object of Exception class type



Base class for all exception

Error message: A human-readable description of the problem.
Name of the exception: This helps identify the specific type of error that occurred.
Stack trace: Information about the sequence of method calls that led to the error. This can be helpful for debugging purposes.

Error message: This is exactly right. It's a clear and concise explanation of the issue, written in a way that a human user (not necessarily a programmer) can understand.
Name of the exception: Spot on! This identifies the specific category of error that occurred. Think of it as a label that helps pinpoint the general problem area.
Stack trace: Your definition is perfect. This provides a detailed breakdown of the sequence of function calls that led to the error. It's like a roadmap that shows you the exact steps that caused the program to go wrong. This is primarily valuable for programmers when debugging code.

Checked Exceptions: Forced error handling (compile-time safety) for potential issues outside the program's control (e.g., file access).
Unchecked Exceptions: The programmer decides on handling (runtime flexibility) for errors likely caused by coding issues (e.g., division by zero).

```
try{
FileInputStream fs = new FileInputStream(“stud.txt”);
}
catch(Exception e){
//error handling code here
}
```

java finally block is a block used to execute important code such as print wether as exception handle or not

the printStackTrace() method in Java is a tool used to handle exceptions and errors it is a method of Java provable class that prints the provable along with other detect line numbers and class names where the exception occurs.

Checked Exception

- IOException
- FileNotFoundException
- ClassNotFoundException
- MalFormaedURLException
- SQLException
- InterruptedException
- EOFException

Unchecked Exception

- ArithmaticException
- NullPointerException
- ArrayIndexOutOfBoundException
- IndexOutOfBoundException
- ClassCastException

Multi catch Before Java 7 we had to catch only one exception type in each catch block so whenever we needed to handle more than one exception but take the same action for all exceptions we had to have more than one catch block containing the same code.

Final re-throw using the final keyword always you to throw an exception of the exact dynamic type that will be thrown. An exception is rethrown when a catch block receives desired ether that it can't process that exception or that it can parshly process it

Employee e1 = new Employee(…);
Class c1=e1.getClass();

Class c2=Clas.forName("Employee");

public Field[] getFields();

public Field[] getDeclaredFields();

public Method[] getMethods();

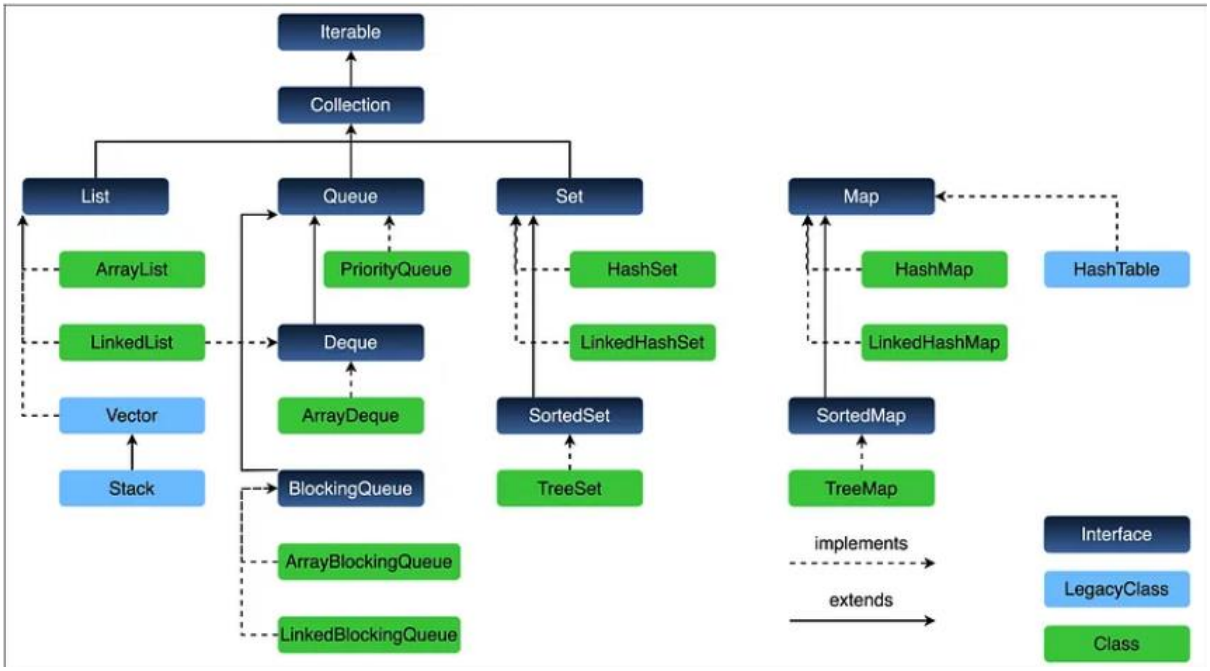
public Method[] getDeclaredMethods();

public Constructor<?>[] getConstructors();

public Constructor<?>[] getDeclaredConstructors();

what is genric
an enum is a special class that represents a group of constant unchanged variables like final variable to create an enum variable insted of class and interface

```
enum Level{
Low,Medium,High
}
```



List in Java provides the facility to maintain the order collection it contains the index-based method to insert update delete and search, it can have duplicate elements also we can also null element in the list.

Set the set as an interface available in the java.util class the set interface extend the collection interface.
Duplicates are not allowed in sets.
Sorted set and navigable set

Map Represent a collection of key-value pair

An iterator in an object that can be used to loop throw collections like ArrayList and HashSet is called an iterator because iterating is the technical term for a loop. package-java.util

A comparator interface is used to order the objects of a user-defined class a comparator object is capable of comparing to objects of the same class

ArrayList java ArrayList class uses a dynamic array for storing the element it is like an array there is no size limit we can add or remove an element at any time and it is more feasible

LinkedList class:- java-util, this class is an implementation of the linked list data structure it is data structure where is data store in non continues

Vector class implement a growable array of objects vector fol in legacy class but now fully compatable in collection

TreeSet contain unique value, collection of sorted and unordered elements, cannot contains null value

A Navigable Set is a sorted set.
Used for searching purposes.
A NavigableSet can be accessed by

A list allows duplicate elements and multiple null element stores, a list is an index sequence, and List implementation is ArrayList, linked list, vector, stack.

Set don't allow duplicate, null element store only one, set non-index sequence, set implementation HashSet, LinkedHashSet

Map Interface
Map associate key with values
The map cannot contain duplicate keys
Key and value be of any Java class or interface type
one key can have at most one value
It is implemented by HashMap and HashTable.

Synchronize in Java is the process that allows only one thread at a particular time too complete the task

Queue Interface
Holding Element prior to Processing
Provide Additional insertion, removal and inspection
Queue FIFO manner but not necessarily
implemented by LinkedList and PriorityQueue

priority queue is also a class is defined in the collection framework that gives a way for processing the object based on priority.

it is already decided that inertion and removal are in the FIFO pattern.

.peek() retrieves the element but doesn't head of 'this' keyword
.poll() retrieve and remove the head of the 'this' keyword.

Why generic
generics are a feature in Java that allows a to write code that works with different data types this can be useful for a variety of reasons like type safety code reuse readability generic means parameter type the idea is to allow type integer string an entity such as class interface and method that operates on parameter type that called generic type.

RTTI:- Run time type information also known as Run time type identification is a feature of several programming languages that makes data about an object data type available at run time.

A comparable java interface is used to order the objects of the user-defined class
the interface is found in Java.lang package and contains only one method compared to the method it provides the single sorting sequence only ex. You can sort the element on the basis of a single data member only it may roll no, name
int compareTo(Object ob);
collections.sort(); and Arrays.sort();

Comparator comparator interface is used to order the objects of a user-defined class. Java.util package

Comparable vs comparator
provide compareTo() method to sort elements, present in java.lang package, we can sort element by collections.sort(list) method, single sorting
provide compare() method to sort element, present in java.util package, we can sort list collections.sort(list, comparator) method, multi sorting

Why we can use the comparator interface
the comparator interface in Java is used to order objects of the class it is used to provide a custom sorting order for a collection of objects the comparator interface is a single method compare which takes to object as an argument and return an integer value and indicates wether the first object is less then equals to or greater then the second object

The comparable interface in Java is used to define a natural order among objects or classes this means that you can use the Comparable interface to sort objects of a class in ascending or descending order

Properties class the properties object contains key and value pair both as a String the java.util.properties class is the subclass of the hash table. It can be used to get property value based on the property class provide a method to get data from the properties file and store data in the property file

Arbitrary object in Java An arbitrary object is an object that can be any type it is not restricted to a specific class or interface. Arbitrary objects are used in a variety of ways such as being passed as an argument to a method or being a method

An instance method in Java is a method that is defined in a class and can only be accessed by the object of that class. Instance method used to act on and they can access and modify the set of the object they belong

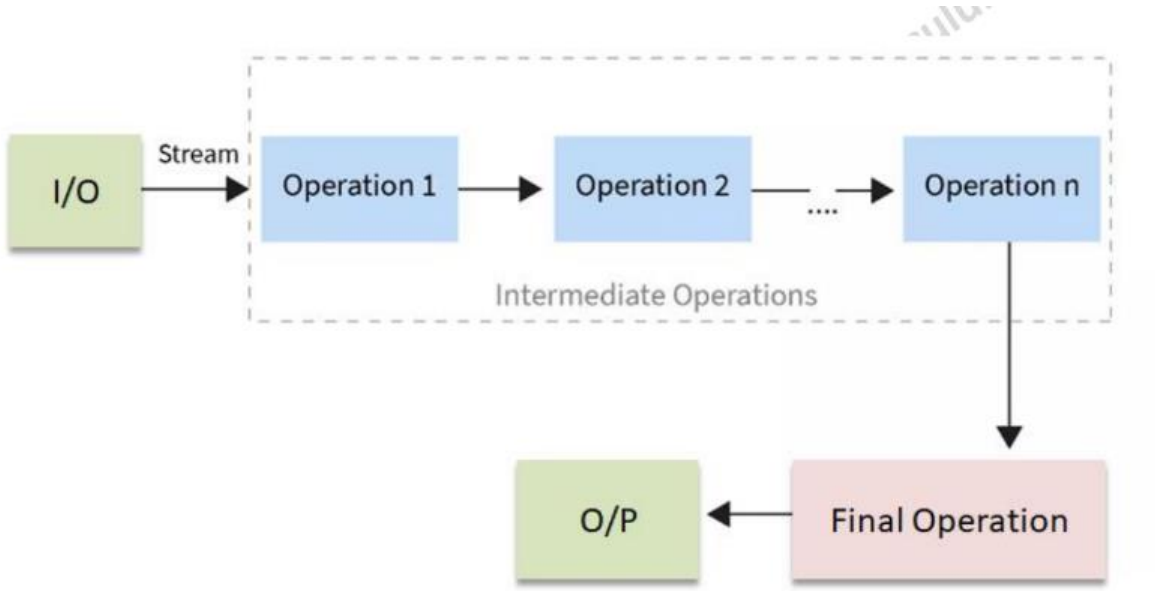
Reference to one instance method of an arbitrary object of a particular type
for ex.
String[] stringArray={“Barble”,” Seed”,}

:: method reference operator..9

Stream

Introduce in Java 8 stream API is used to process a collection of objects a stream in Java is a sequence of an object that support various methods which can be a pipeline to produce the desired result

Use of Stream
Stream API is a way to express and process the collection of object
A stream in Java enables us to perform us to like filtering, mapping reducing and sorting



Characteristics
chain-together, intermediate operation, transforms a stream into another stream, it enables the concept of filtering where one method filters data and passes it to another method after the process

.filter():- we can use the logical operator in Java to unite multiple conditions in the filter method, two conditions in the filter method are length using the “and” logical operator
.map():- Stream map returns a stream consisting of the result of applying the given function to the element of the stream

Designed for processing data
Not used to store data

Functional-style operation
like map, filter and reduce enable developers to express data processing in a declarative and readable way.

Pipelining allows for chaining multiple operations together to form a pipeline

Lazy evaluation
- Intermediate operations are not executed until the final operation is invoked
- Improves efficiency

Parallel processing
- Divide the dataset into chunks and process them concurrently

Can be easily output as an array or list
not reusable

Collections hold data value as the data structure
Stream arrays or collections are computed on demand

Adjusters are for modifying termination objects they exist to externalize the process of adjustment

Permitting different operates as per the strategy

Terminal operation
forEach()
Collect()
match()

Multithreading

Multiprocessor:- A system that is built using multiple processors for performing tasks.

MultiTask:- An approach where multiple tasks get executed by one or more processors

Multiprocess:- A concrete implementation of tasking

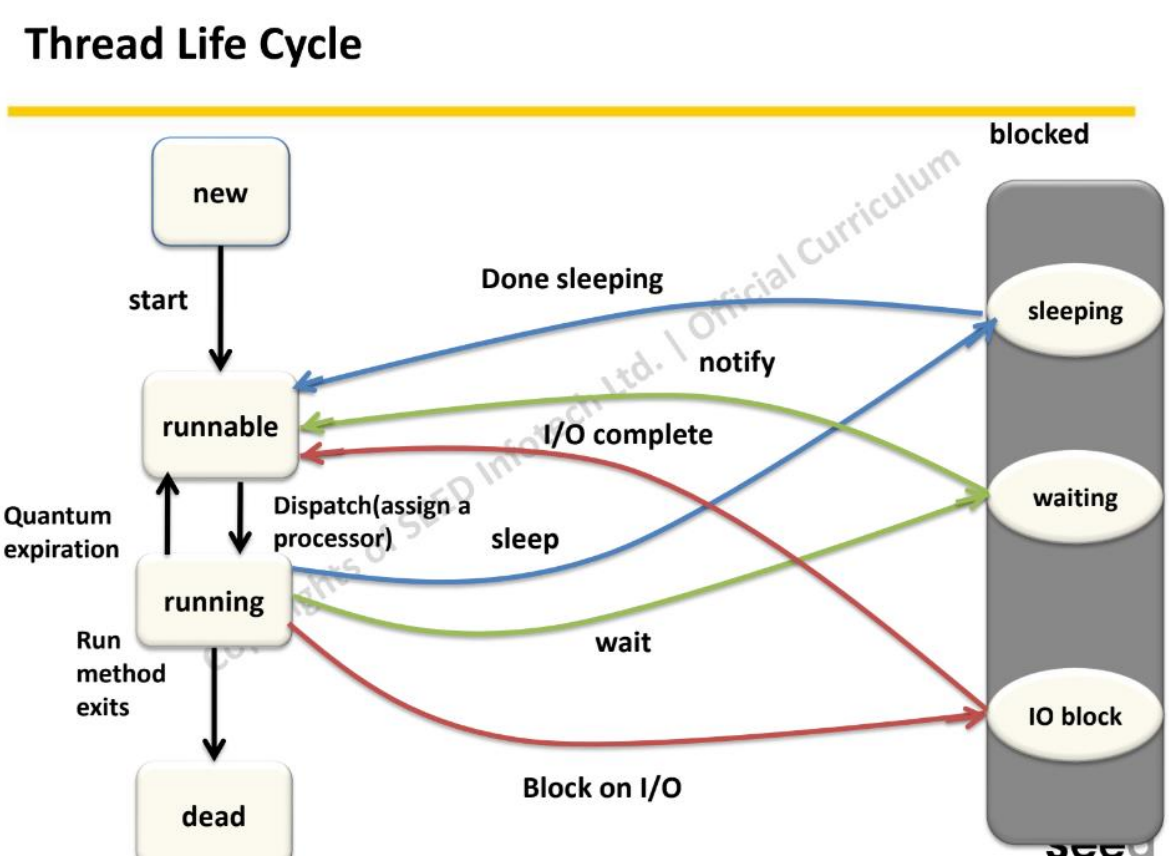
MultiThread:- An approach where a single process is split into multiple execution thread

Threads

- **Lightweight**
  - Threads are lightweight as they share the same memory space.
- **Single Task Allocation**
  - Each thread is responsible for a single task within a process.
- **For Business Rules**
  - Typically used to execute business rules and fine-grained tasks within an application.
- **Inter-Thread Communication**
  - Faster communication as threads share the same memory and resources.

Processes

- **Heavyweight**
  - Processes are heavyweight as they have separate memory spaces.
- **Collection of Tasks**
  - A process can contain multiple threads, each performing different tasks.
- **For Application Logic**
  - Used for executing broader application logic and tasks that require isolation.
- **Inter-Process Communication**
  - Slower communication due to separate memory spaces, often requiring IPC mechanisms like pipes, sockets, or shared memory.



Now whenever a new thread is created it is always in the new state(the code has not been run)

Active state when a thread invokes the start method it moves from nest to active state

The active state content 2 state is runnable and running

A thread that is ready to run is then moved to the runnable state in the runnable state the thread may be running or ready to run at any given instance of time it is the duty of the thread scheduler to provide the thread time to run

When the thread get to cpu it moves from the runnable to running stage generally the most common change in the state of change is from runnable to running and again back to running

Block or waite

Terminator when the thread completes task

Pre-emptive In pre-emptive scheduler the scheduler will execute the project for a specific amount of time before passing it on to the next process in line it as to with to be excuated again by the scheduler the process may go to the ready state to running state or the waiting state to ready state

Non-Per-preemptive scheduling is a method that may be used when a process terminates or switches,

Running to a waiting state when processer is assigned to the process they keep the process until it is eliminated or reaches a waiting state

Thread priority
to decide by thread scheduler when each thread should be allowed to run

Sync in java is the capacity to control the access of multiple threads to any shared resource

Java sync is a better operation where you allow one thread to access the shared resources

Sync block is used to perform synchronization on any specific resource of the method suppose we have 50 line of code in our method but we want to synchronize only 5 line that time we use sync block

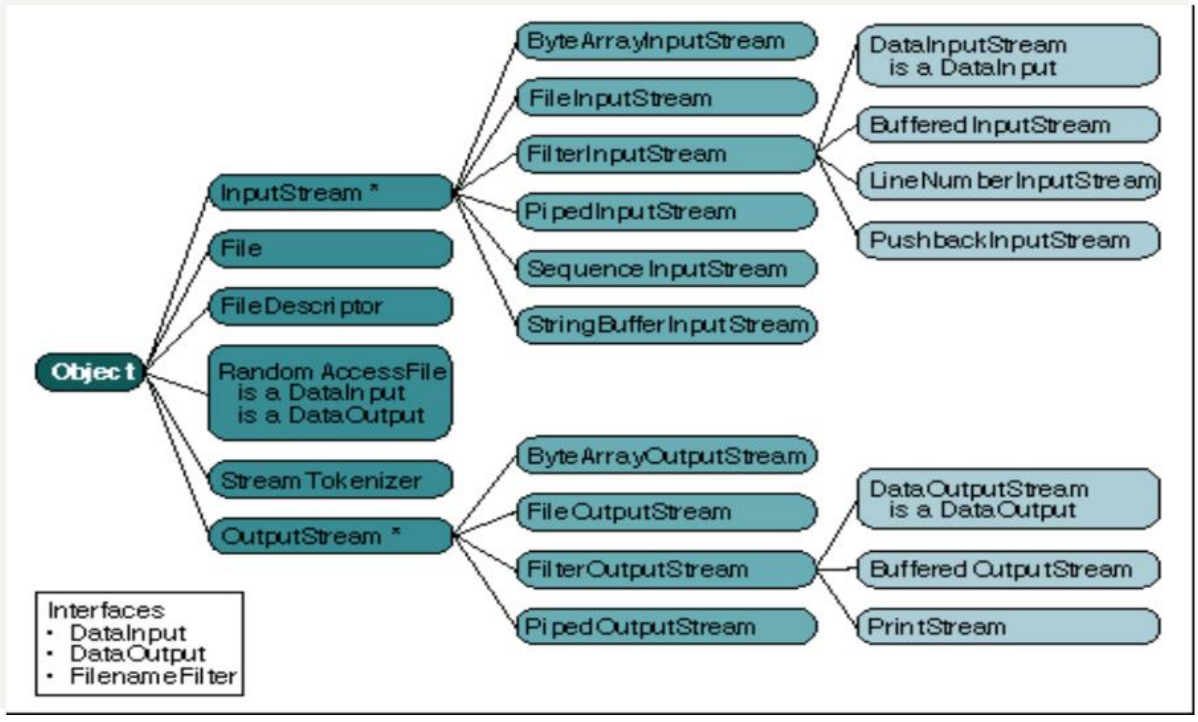
The wait() method is related to the object class the wait method is responsible for sending the calling thread into the waiting state. The thread remain in waiting state until another thread doesn't invokes the notify or notify all method for the object the thread resumes the execution to obtain ownership of the monitor

Atomic access in the context of java atomic operation are those that can be perform automaticly without interfrace of other thread

Monitor object is a pattern that control the concurrent access of a method in an object if there are some concurrent thread calling method at same time only one thread can execute this method

Stream API is used to process collections of objects a stream in java is a sequence of objects that support various method which can be pipeline to to produce the desired result

Use of stream in java
stream API is a way to express and process collections of object
enable us to perform operation, light, filtering, mapping, reducing and sorting



For taking input from the user we can use Scanner

boolean canRead(),
boolean isAbsolute(),
boolean exists(),
long length(),
boolean delete(),
boolean createNewFile(),
boolean mkdir()

an interface that doesn’t content method fields and constant is known as a marker interface in other words an empty interface is known as a marker interface the serializable and clonable interfaces are an examples of marker interfaces