Threads:

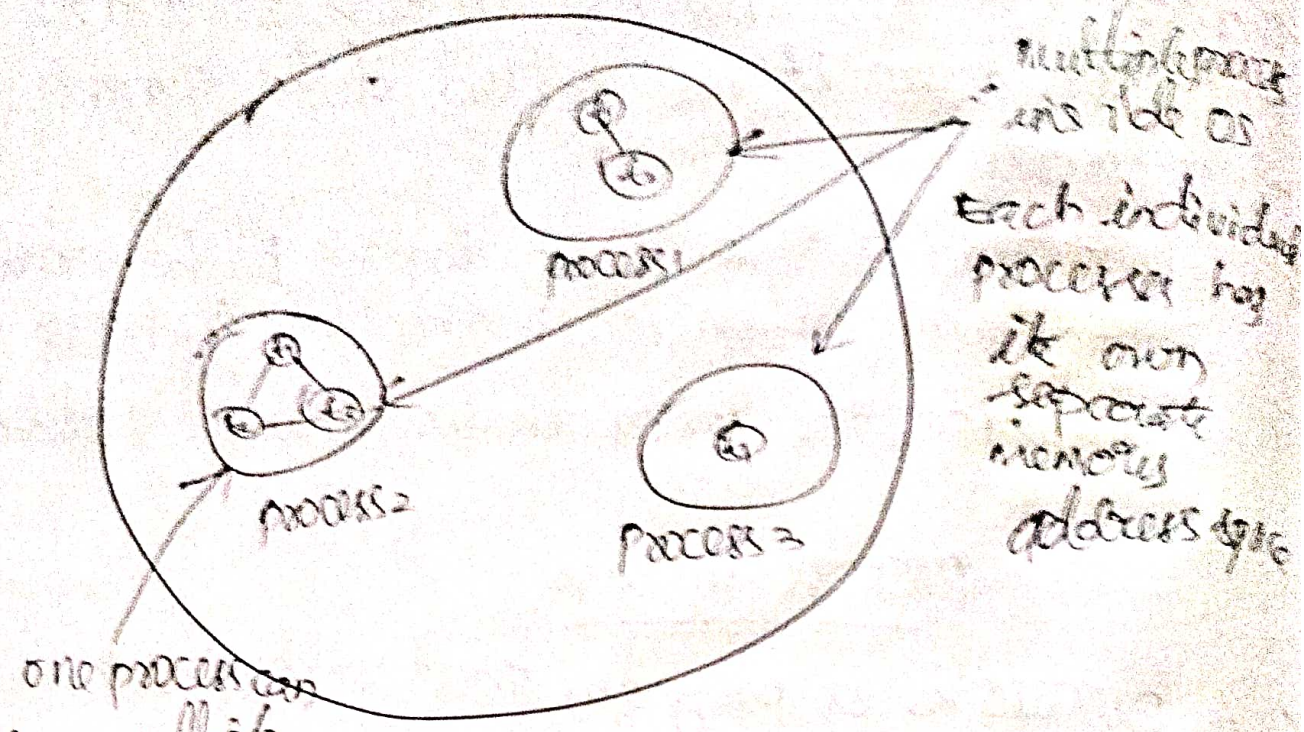
27/6/24

* A Thread in Java simply represents a single independent path of execution of a group of statements.
* There is always at least one thread running internally in every program and this thread is used by JVM to execute statements in the program.
* When a program contains a single flow of control, it is called single-threaded program.
* A light weight process which runs under resources of main process
* Every Java program has always at least one thread, even if you do not create any thread. This thread is called main Thread.

* The main thread is also called parent thread and the rest of threads that are generated from it are called child threads of the program.
* Main thread is the last thread to be created in a program. When main thread finishes the execution, the program terminates immediately

process in Java

* A process is a program that executes as a single thread.
* When we will create a new thread in a program it shares the same memory address space with other threads in a program whereas every individual process has its own separate memory address space.
* Creating a thread takes fewer resources than creating a new process.
* Each process can have more than one thread.
* Every individual process has its own separate memory address space and can execute a different program.



Difference between Thread and Process

process	Thread
* process is heavy weight or resource intensive	* Thread is light weight taking lesser resources than a process
* process switching needs interaction with operating system	* Thread switching does not need to interact with operating system.
* In multiple processing environments each process executes the same code but has its own memory and file resources	* All threads can share same set of open file, child processes.
* Multiple process without using threads use more resources	* Multiple threaded process use fewer resources.

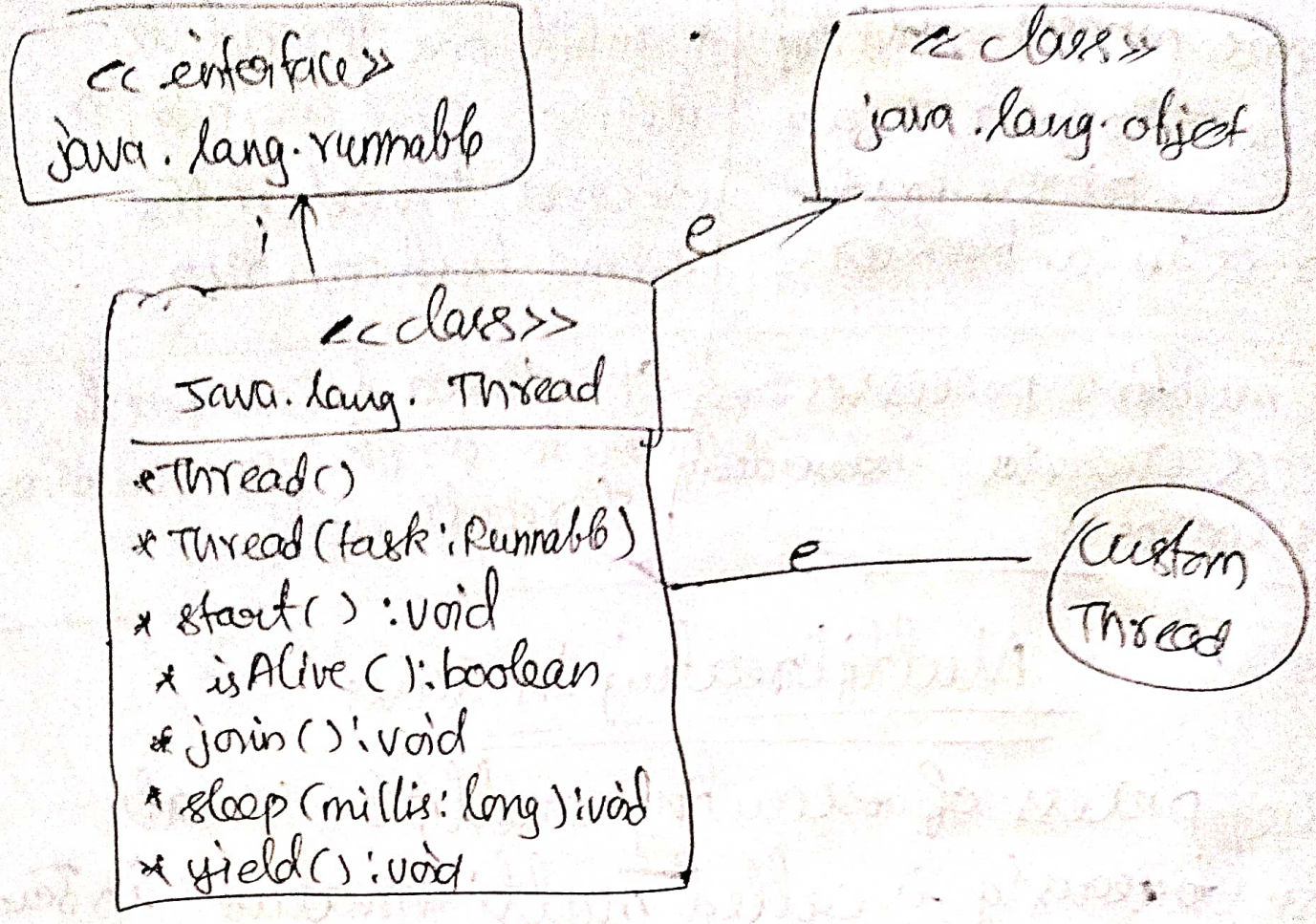
* if one process is blocked then no other process can execute until the first process is unblocked	* while one thread is blocked and waiting, a second thread in the same task can run
* In multiple processes each process operates independently of the others	One thread can read, write or change another thread's data

Multithreading in Java

* The process of executing multiple threads simultaneously is called multithreading in Java
* Multithreading is a technique or programming concept in which a program (process) is divided into two or more subprograms (subprocess), each of which can perform different tasks simultaneously
* When a program contains more than on thread, the cpu can switch between two threads to execute them at the same time.
* The switching between two threads is known as context switch.

Thread class

* Thread class contains several constructors for creating threads for tasks and methods for controlling threads.
* It is a predefined class declared in java.lang default package.



Thread(): This is a basic and default constructor without parameters. It simply creates an object of Thread class.

Thread(String name): It is create a new Thread object with specified name to a thread.

Thread(Runnable r): It creates a thread object ~~with specified name to a thread.~~ by passing a parameter r as a reference to an object of the class that implements Runnable interface.

Thread(Runnable r, String name): This constructor creates a thread object by passing two arguments r and name. Here r is a reference of an object of class that implements Runnable interface.

Creation of Threads

There are two ways to create a new thread in Java
* one is by extending java.lang.Thread class:
* create a class that extends the Thread class.
* in order to extend a thread, we will use a keyword extends. The Threads class is found in Java.lang package.

Syntax: class Myclass extends Thread

* Another is by implementing java.lang.Runnable interface.
* create a class that implements the runnable interface
* Class to implement only the run method that constitutes the new Thread.