PW SKILLS

Lecture

MultiThreading

List of Concepts Involved:

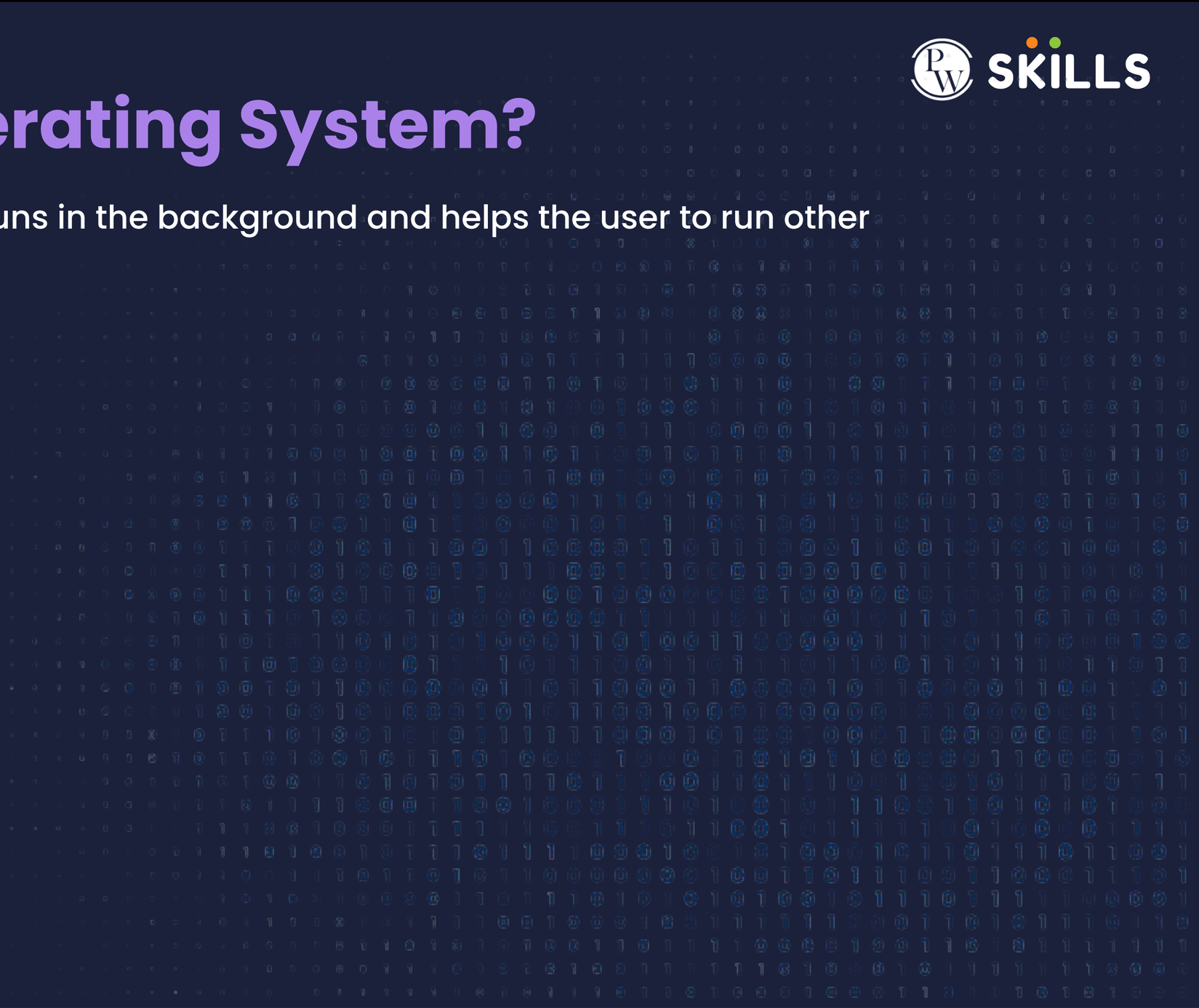
- What is an Operating System?
- Tasking vs Multitasking
- Process
- Threads
- How to create Threads
- run() method
- Multiple task within single run()
- Different states of Thread
- synchronised keyword
- Deadlock
- Producer - Consumer problem

Topics covered in Previous Session:

- Exception Handling in Java

What is an Operating System?

It is a system software which runs in the background and helps the user to run other applications.



MultiTasking

Executing several tasks simultaneously is the concept of multitasking.

There are 2 types of Multitasking.

a. Process based multitasking

b. Thread based multitasking.

Process based multitasking

Executing several tasks simultaneously where each task is a separate independent process such type of multitasking is called **"process based multitasking"**.

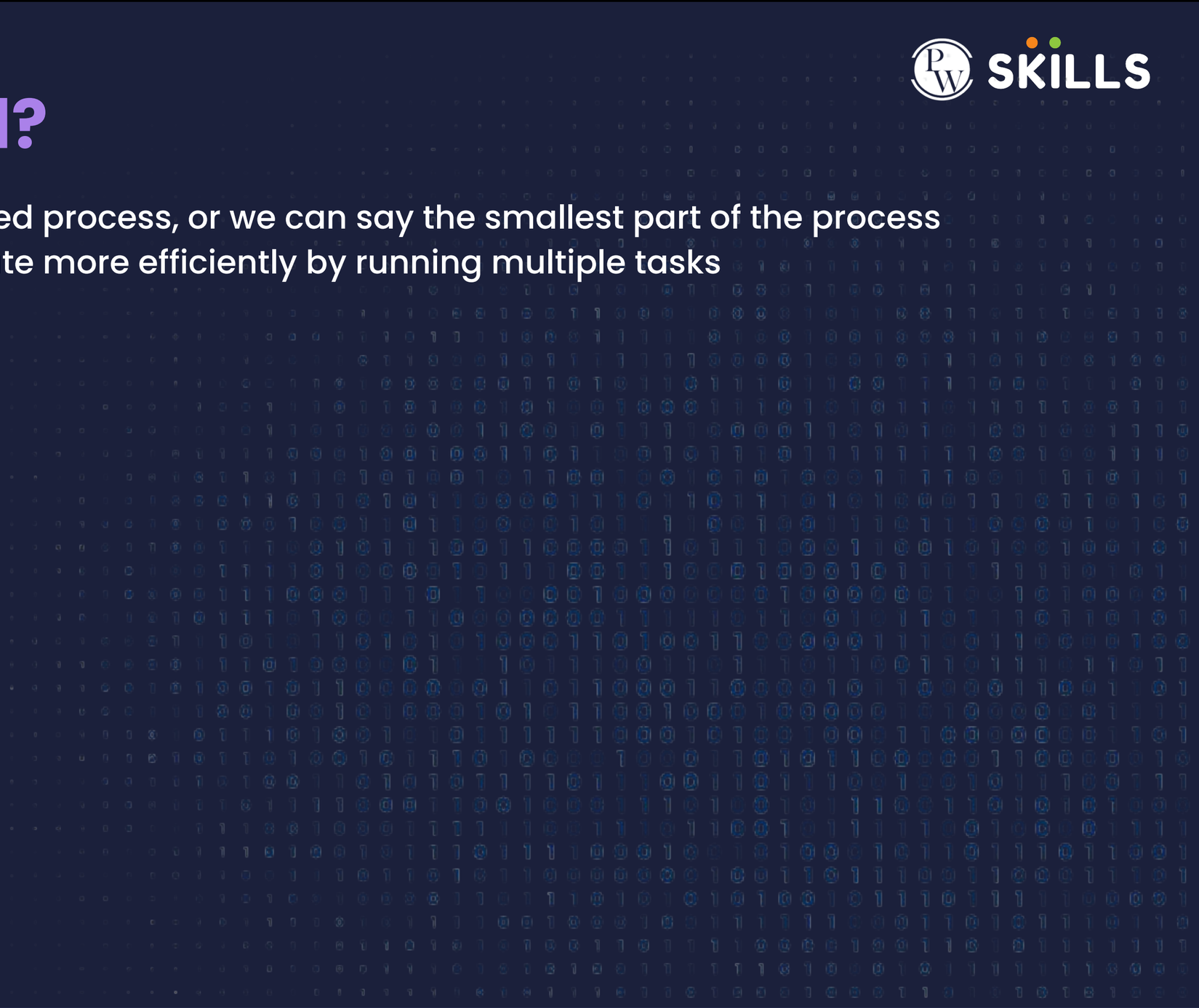
Thread based multitasking

Executing several tasks simultaneously where each task is a separate independent part of the same Program, is called **"Thread based MultiTasking"**.

Each independent part is called a "Thread".

What is Thread?

A Thread is a very light-weighted process, or we can say the smallest part of the process that allows a program to operate more efficiently by running multiple tasks simultaneously.



How to create Threads

There are two ways to create a thread:

1. By extending Thread class
2. By implementing Runnable interface.

Thread Scheduler

- If multiple threads are waiting to execute,then which thread will execute 1st is decided by Thread Scheduler which is part of JVM.

- In the case of MultiThreading we can't predict the exact output, only possible output we can expect.

run() method

The run() method is available in the thread class constructed using a separate Runnable object. Otherwise, this method does nothing and returns. We can call the run() method multiple times.

The run() method can be called in two ways which are as follows:

1. Using the start() method.
2. Using the run() method itself.

Diff b/w start() and run()

- If we call start() and seperate thread will be created which is responsible to execute the run() method.

- If we call run(), no separate thread will be created, rather the method will be called just like a normal method by main thread.

Importance of Thread class start() method

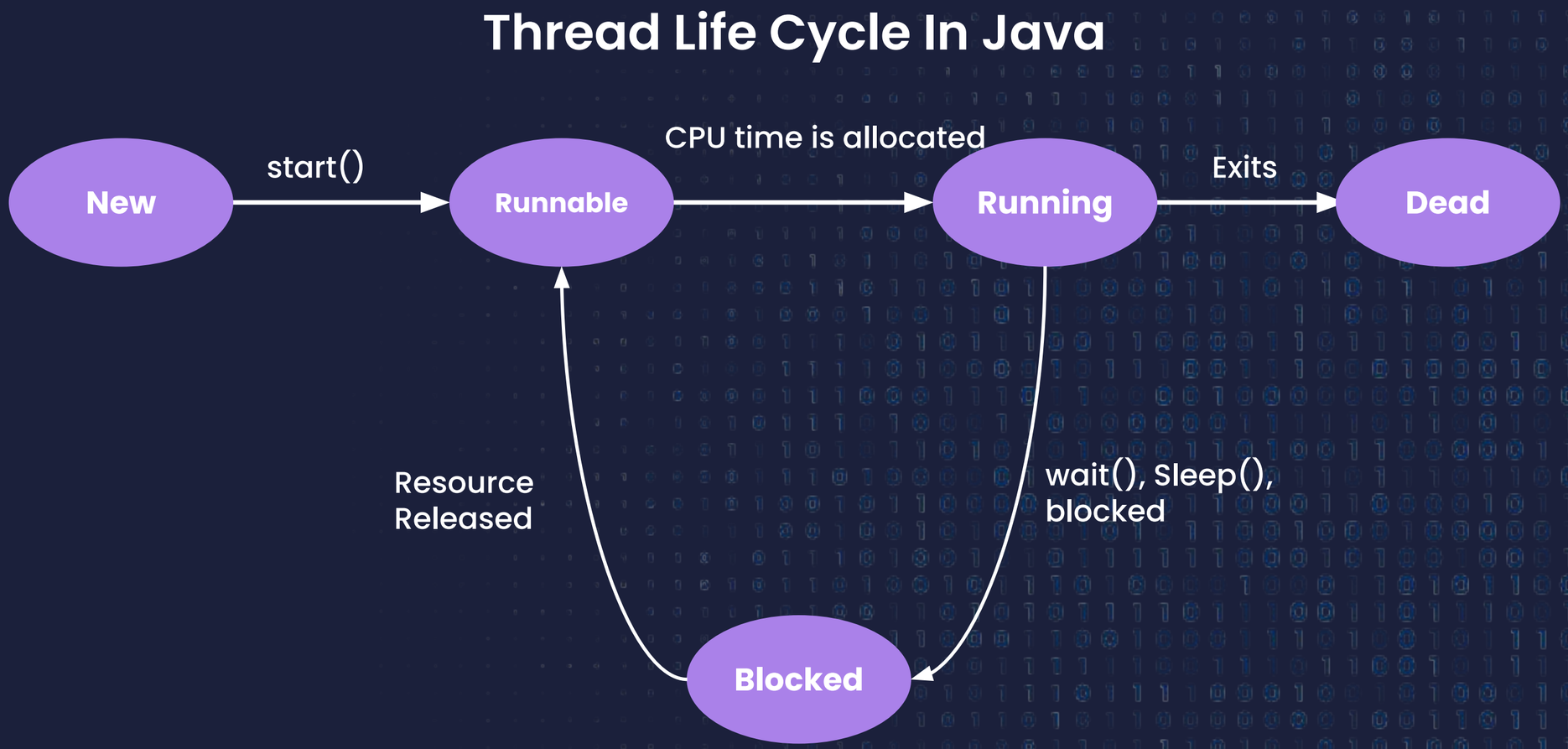
- For every thread, required mandatory activities like registering the thread with Thread Scheduler will be taken care by Thread class
- start() method and the programmer is responsible for just doing the job of the Thread inside run() method.
- start() acts like an assistance to programmers.
- start()
  {
  register thread with ThreadScheduler
  All other mandatory low level activities
  invoke or call the run() method.
  }
- We can conclude that without executing the Thread class start() method there is no chance of starting a new Thread in java.
- Due to this, start() is considered the **heart of MultiThreading.**

Different states of Thread

A thread is a path of execution in a program that goes through the following states of a thread. The five states are as follows:

1. New
2. Runnable
3. Running
4. Blocked (Non-runnable state)
5. Dead

Life cycle of a Thread



synchronized keyword

- synchronized keyword is used to make the class or method thread-safe which means only one thread can have lock of synchronized method and use it, other threads have to wait till the lock releases and anyone of them acquire that lock.

- It is important to use if our program is running in multi-threaded environment where two or more threads execute simultaneously. But sometimes it also causes a problem which is called Deadlock.

Deadlock

Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.

Here, both threads are waiting for each other to unlock resources R1 and R2, but thread cannot release lock for resource R2 until it gets hold of resource R1.

Fig: Example of deadlock condition

Inter Thread Communication

Two threads should interact with each other, how?

eg: Producer Consumer Problem

ProducerEnd

- Producer duty is to produce the data and once the data is produced update the variable called "DataProvider" to true
- This action should be done by "Producer Thread"

Producer Consumer Problem

Consumer End

- Consumer Thread should consume the data produced by the Producer
- Consumer Thread should check the data provider status,if it is true consume the data otherwise sleep for some time and again check for the data provider status.

Methods

To reduce the efficiency problem we use the methods given by Object class

1. wait()
2. notify()
3. notifyAll()

Next Lecture

- Collection Framework

PW SKILLS

THANK YOU