MULTITHREADING ASSIGNMENT

1. WHY WE USE MULTITHREADING?

Multithreading allows the execution of multiple parts of a program at the same time. These parts are known as threads and are lightweight processes available within the process. So multithreading leads to maximum utilization of the CPU by multitasking.

1. What do you understand by multi-processing?

**multiprocessing**, in computing, a mode of operation in which two or more processors in a [computer](https://www.britannica.com/technology/computer) simultaneously process two or more different portions of the same [program](https://www.britannica.com/technology/computer-program) (set of instructions). Multiprocessing is typically carried out by two or more [microprocessors](https://www.britannica.com/technology/microprocessor), each of which is in effect a [central processing unit](https://www.britannica.com/technology/central-processing-unit).

1. Diff. between multitasking, multithreading and multi-processing

Multiprogramming

* "The concurrent residency of more than one program in the main memory is referred as multiprogramming."
* Since multiple programs are resident in the memory, as soon as the currently executing program finishes its execution, the next program is dispatched for its consumption.
* Also if the currently executing program asks for input output resources then meanwhile another program is dispatched to the CPU for execution.
* The main objective of multiprogramming is:
  + Maximum CPU utilization.
  + Efficient management of the main memory.

Multiprocessing

* When one system is connected to more than one processor which collectively work for the completion of the task, it is called as multiprocessing systems.
* Multiprocessing systems can be divided in two types:
  + **Symmetric Multiprocessing:** The operating system here resides on one processor and the other processors run user's programs.
  + **Asymmetric Multiprocessing:** The OS runs on any available processor or all the processor simultaneously run the user program.

Multithreading

* "Multithreading is a conceptual programming paradigm where a process is divided into a number of sub-processes called as threads. Each thread is independent and has its own path of execution with enabled inter thread communication."
* "Thread is the path followed while executing a program. Each thread has its own program counter, stack and register."
* A thread is a light weight process.

Multitasking

* Earlier when computers were invented, a user was allowed to submit only job or task at a time. But later with availability of high-speed processor, one can submit more than one task.
* So the capability of OS to accept more the one task per user is termed as multitasking.
* Multiple jobs are executed by the CPU simultaneously by switching between them.
* The various job can be accepted from same user or different users. There are 2 types of multitasking systems:
  + **Single User Multitasking**
  + **Multi User multitasking**

1. Explore the location of the start() method

The **java.lang.Thread.start()**method causes this thread to begin execution, the Java Virtual Machine calls the run method of this thread.

1. Explore garbage collector (GC)

Garbage Collection is process of reclaiming the runtime unused memory automatically. In other words, it is a way to destroy the unused objects.

### **Advantage of Garbage Collection**

* It makes java **memory efficient** because garbage collector removes the unreferenced objects from heap memory.
* It is **automatically done** by the garbage collector(a part of JVM) so we don't need to make extra efforts.

## How can an object be unreferenced?

## There are many ways:

* By nulling the reference
* By assigning a reference to another
* By anonymous object etc.

1. Explore max, min and norm priority and their usage

The **setPriority()** method of thread class is used to change the thread's priority. Every thread has a priority which is represented by the integer number between 1 to 10.

Thread class provides 3 constant properties:

1. **public static int MIN\_PRIORITY:** It is the maximum priority of a thread. The value of it is 1.
2. **public static int NORM\_PRIORITY:** It is the normal priority of a thread. The value of it is 5.
3. **public static int MAX\_PRIORITY:** It is the minimum priority of a thread. The value of it is 10.

We can also set the priority of thread between 1 to 10. This priority is known as custom priority or user defined priority.