**Multithreading – Part-04 – Thread Priorities**

* **Priority:**

Every Thread in Java has priority. It may be default priority generated by JVM or customized priority provided by programmer.

The valid range of thread priorities is 1 to 10.

Where 1 is min priority and 10 is max priority.

Thread defines the following constants to represent some standard priorities.

Thread.MIN\_PRIORITY – 1

Thread.NORM\_PRIORITY – 5

Thread.MAX\_PRIORITY – 10

On the following priorities which are valid one?

0 // Invalid

1 // Valid

10 // Valid

Thread.LOW\_PRIORITY // Invalid

Thread.HIGH\_PRIORITY // Invalid

Thread.MIN\_PRIORITY // Valid

Thread.NORM\_PRIORITY // Valid

Thread scheduler will use priorities while allocating processor.

The thread which is having highest priority will get the chance first.

If two threads have same priority, then we can’t expect exact execution order. It depends on thread scheduler.

Note: Recall the traffic singal priority among, Minister, CM, PM, President.

* **How to get and set thread priority:**

Thread class defines the following methods to get and set priority of a thread.

public final int getPriority();

public final void setPriority(int p);

Allowed values range of p is 1 to 10.

If we set morethan this, we will get Runtime Exception IllegalArgumentException.

Example:

t.setPriority(7); // Valid

t.setPriority(17); // IllegalArgumentException

* **Default Priority:**

The default priority only for the main thread is 5, but for all remaining threads default priority will be inherited from parent to child. That is whatever priority parent thread has the same priority will be there for the child thread.

Example:

class MyThread extends Thread{

}

class Test{

public static void main(String[] args){

System.out.println(Thread.currentThread().getName());

Thread.currentThread().setPriority(15);

Thread.currentThread().setPriority(7); // Iine-1

MyThread t = mew MyThread();

System.out.println(t.getPriority());

}

}

If we comment line-1 then child thread priority will become 5

Note:

Parent class is different and parent thread is different.

class MyThread extends Thread

For the above class declaration:

Parent class is Thread

Parent Thread is main thread.

Main thread is the one who creates the object of the MyThread.

* **Priority Example:**

class MyThread extends Thread{

public void run(){

for(int i = 0; i < 10; i++){

System.out.println(“Child Thread”);

}

}

}

class ThreadPrioritiesDemo{

public static void main(String[] args){

MyThread t = new MyThread();

t.setPriority(10); // line -1

t.start();

for(int i = 0; i < 10; i++){

System.out.println(“Main Thread”);

}

}

}

Note:

If we are commenting line-1, then both main and child threads have the same priority 5 and hence we can’t expect the execution order and exact output.

If we are not commenting line-1, then main thread has a priority 5 and child thread has the priority 10 hence child thread will get the chance first followed by main thread. In this case output is.

Child Thread, Child Thread 10 times. Followed by

Main Thread, Main Thread 10 times.

Some platforms won’t provide proper support for thread priorities, because of this even though we set the priorty we can’t expect the correct output.