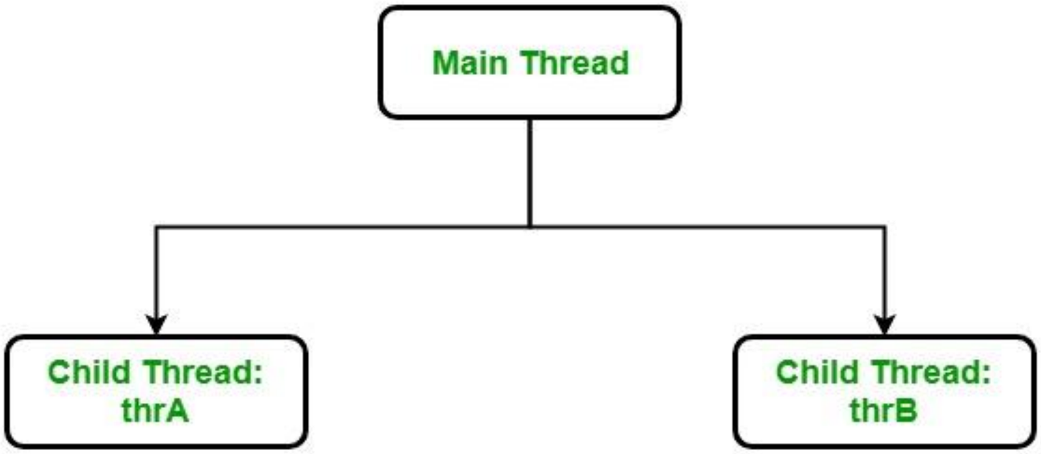
El-Shorouk Academy
Higher Institute for Computer
&Term Information Technology
DR. Negm Eldin Shawky

Acad. Year: 2023/2024
Term: 1st
Year: 4
Computer Science Department

Network Programming

Section 9

- A process is a program in execution. process may be divided into several independent units known as threads.
- A thread may be assumed as a subset of a process.
- A program with master thread and children’s threads. This is called multithread.



- ➢ **Simple program creating and invoking a thread object by extending the standard Thread class.**

```
class ThreadA extends Thread {
public void run() {
for (int i = 1; i <= 5; i++) {
System.out.println("Thread Count= " + i);

try {
Thread.sleep(1000);  // Sleep for 1 second
} catch (InterruptedException ex) {
ex.printStackTrace();

}

}
}
    }
public class ThreadTest {

   public static void main(String args[]) {
      // Create and start a new thread
      ThreadA thread1 = new ThreadA();
      thread1.start();

      // Create and start another thread
      ThreadA thread2 = new ThreadA();
      thread2.start();
   }
    }
```

- **Simple program creating and invoking a thread object by implementing Runnable interface.**

```
   class MyThread1 implements Runnable {
   public void run() {
      // Code to be executed by the thread
      for (int i = 1; i <= 5; i++) {
         System.out.println("Thread Count: " + i);
         try {
            Thread.sleep(1000);  // Sleep for 1 second
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
   }
}

public class ThreadTest {

   public static void main(String[] args) {
      // Create an instance of the Runnable implementation
      MyThread1 myRunnable = new MyThread1();

      // Create a thread using the Runnable instance
      Thread thread = new Thread(myRunnable);

      // Start the thread
      thread.start();
```

```
    }
}
```

- **A java program with multiple threads and Thread Priority.**

```
 class A extends Thread {

    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println("Thread  Count " + i);
            try {
                Thread.sleep(100);  // Sleep for 100 milliseconds
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
class B extends Thread {

    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println("Thread  Count " + i);
            try {
                Thread.sleep(100);  // Sleep for 100 milliseconds
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class C extends Thread {

    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println("Thread  Count " + i);
            try {
                Thread.sleep(100);  // Sleep for 100 milliseconds
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
public class ThreadTest {

    public static void main(String[] args) {
        // Create instances of MyThread with different priorities
        A thread1 = new A();
        A thread2 = new A();
        A thread3 = new A();

        thread1.setPriority(Thread.MIN_PRIORITY);
        thread2.setPriority(thread2.getPriority() + 1);
        thread3.setPriority(Thread.MAX_PRIORITY);

        // Start the threads
        thread1.start();
        thread2.start();
        thread3.start();
    }
}
```