1. Create new thread class by extending the class **Thread**.

```
class RaceCar extends Thread
{
   int finish;
   String name;

   RaceCar(int finish, String name)
   {
      this.finish = finish;
      this.name = name;
   }

   public void run()
   {
      for (int i=0; i<finish; i++)
      {
         System.out.println(name + ": " + (i+1));

         try {
            Thread.sleep(Math.round(Math.random()*5000));
         } catch(Exception e){}
      }

      System.out.println(name + " finished");
   }

}


public class Race
{
   public static void main(String a[])
   {
      RaceCar[] cars = new RaceCar[5];
      cars[0] = new RaceCar(10, "Mario");
      cars[1] = new RaceCar(10, "Songuku");
      cars[2] = new RaceCar(10, "Herman");
      cars[3] = new RaceCar(10, "Doremon");
      cars[4] = new RaceCar(10, "Hoang Phi Hong");

      for(int i=0; i<5; i++)
         cars[i].start();
   }
}
```

2. Create new thread class by implementing the interface **Runnable**.

```
class Mosquito implements Runnable
{
   private int id;
   public Mosquito(int id)
   {
      this.id = id;
   }

   // override
   public void run()
   {
      if (id == 12) {
         try {
            Thread.sleep(10000);
         } catch(Exception e){}
      }

      for (int i=0; i<10; i++)
      {
         System.out.println("Vo ve " + id);
      }
   }
}

public class RunnableMosquitoTest
{
   public static void main(String args[])
   {
      Mosquito m1 = new Mosquito(12);
      Mosquito m2 = new Mosquito(21);
      Mosquito m3 = new Mosquito(22);
      Mosquito m4 = new Mosquito(32);

      Thread t1 = new Thread(m1);
      t1.start();

      Thread t2 = new Thread(m2);
      t2.start();

      System.out.println(t1.getName());
      System.out.println(t2.getName());
      System.out.println("No. of Thread: " +
Thread.activeCount());
   }
}
```

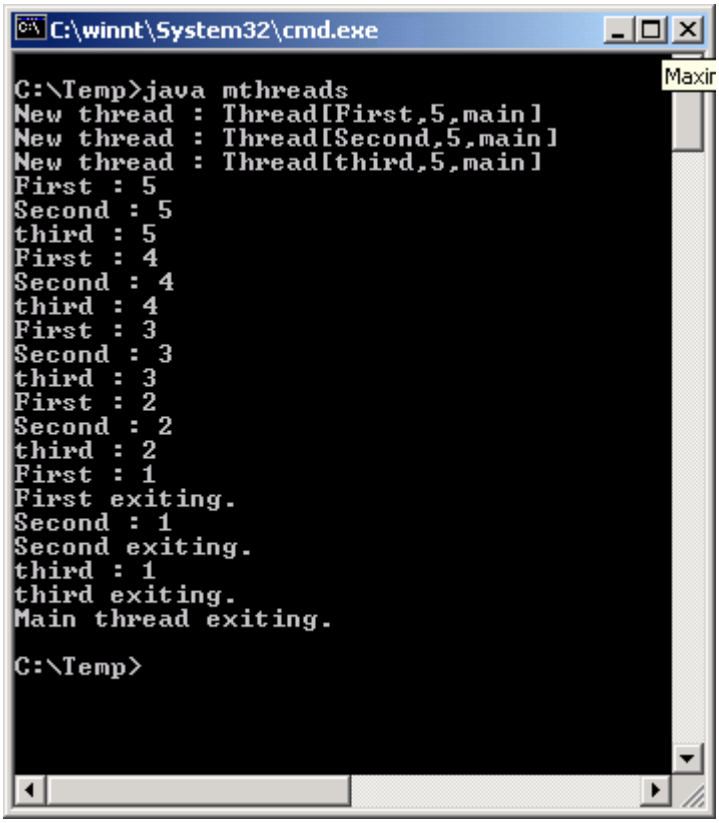
Do It Yourself

1.1. Do workshop of the module 1

1.2. Create a Thread named MyThread (by extending from the class Thread or by implementing the interface Runnable) does these tasks:

- Display name of the current thread.
- Rename the created thread to myJavaThread and display this name.
- Overwrite the method run() by displaying first 10 even numer. Delay time between each display time is 1500 ms.

1.3. Write a program as the following test run:



Create three threads and the main thread. Execute each thread as simultaneous tasks. Display information when exiting each thread.

1.4. Create a new thread class that prints a value in a time duration. Information about a thread includes: message, tim duration, priority.

```
MESSAGE BOARD
====================
Number of messages: 2
```

```
Message 1: multithreading
Timeout: 1000
Priority: high
Message 2: multitasking
Timeout: 2000
Priority: medium

Result:
Multithreading
Multithreading
Multitasking
Multithreading
Multithreading
Multitasking
…..
```

References

+ Java tutorials

+ Javadoc

+ Java2s.com

+ Javapassion.com

+ Java almanac
http://www.exampledepot.com