11. Write a program to illustrate creation of threads using runnable class. (start method start each of the newly created thread. Inside the run method there is sleep() for suspend the thread for 500 milliseconds).

```
class Thread1 implements Runnable
{
   String name;
   Thread t;
   Thread1(String name)
   {
      t=new Thread(this,name);
   }
   public void run()
   {
      System.out.println(t.getName()+"  started");
      try{Thread.sleep(500);}
      catch(InterruptedException e){ };
      System.out.println(t.getName()+"  ended");
   }
}
class Main {
   public static void main(String[] args)  throws InterruptedException {
     Thread1 t1=new Thread1("first");
     Thread1 t2=new Thread1("second");
     t1.t.start();
     t2.t.start();
     System.out.println("Main started..");
    t1.t.join();
    t2.t.join();
     System.out.println("Main ended..");
   }
}
```

OUTPUT 1

Main started..

first started

second started

first ended

second ended

Main ended..

OUTPUT 2

first started

Main started..

second started

first ended

second ended

Main ended..

Observe each time the different order in threads starting and ending.

12. Develop a program to create a class MyThread in this class a constructor, call the base class constructor, using super and start the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed

```
public class Main extends Thread
{
  String name;
  Main(String name)
  {
    super(name);
  }
  public void run()
  {
    System.out.println(getName()+" started");
    try{Thread.sleep(500);}
    catch(InterruptedException e){ };
    System.out.println(getName()+" ended");
  }
```

```
  public static void main(String[] args)  throws InterruptedException {
    Main t1=new Main("first");
    Main t2=new Main("second");
    t1.start();
    t2.start();
    System.out.println("Main started..");
   t1.join();
   t2.join();
    System.out.println("Main ended..");
  }
```

OUTPUT1

```
Main started..
first  started
second  started
second  ended
first  ended
Main ended..
```

OUTPUT 2

```
Main started..
first  started
second  started
first  ended
second  ended
Main ended..
```