

## How to set up a Threaded Animation

### File 1: **MyCreation.java**

Create a Console object

Send it into each class constructor so everything displays on same window

Create a method to construct and start each ObjectClass

```
public static void drawBoat( )
{
    DrawBoat d = new DrawBoat (c);
    d.start ( );
}
```

Main method - run other methods in order.

### File 2 - ?: **ObjectClass.java**

**import** java.lang.\*; // to access Thread class

**public class** ObjectName **extends** Thread

```
{
    static Console c;

    // set up your methods/variables to draw whatever objects you want

    public ObjectName (Console con)
    {
        c = con;
    }

    public void run( )
    {
        // call methods in run
    }
}
```

### Animation Control

If you want one animation to finish before a new one begins - you need to add this inside the MyCreation method where the object starts (the one you want to run from beginning to end)

```
try
{
    ObjectClass.join ();
}
catch (Exception e)
{
}
```

## How to set up a Runnable Threaded Animation

### File 1: MyCreation.java

Create a Console object

Send it into each class constructor so everything displays on same window

Create a method to construct and start each ObjectClass

```
public void drawBoat( )
{
    DrawBoat d = new DrawBoat (c);
    d.run ( );
}
```

Main method - run other methods in order.

### File 2 - ?: ObjectClass.java

**import** java.lang.\*; // to access Thread class

**public class** ObjectName **implements** Runnable

```
{
    Console c;

    // set up your methods/variables to draw whatever objects you want

    public ObjectName (Console con)
    {
        c = con;
    }

    public void run( )
    {
        // call methods in run
    }
}
```

### Animation Control

If you want one animation to finish before a new one begins - you need to add this inside the MyCreation method where the object starts (the one you want to run from beginning to end)

```
try
{
    ObjectClass.join ();
}
catch (Exception e)
{
}
```