A dead lock happens when we have multiple threads on each others resources
Here we have first resource class A, which has two methods which are both synchronized
Similarly we have second resource class B, which has two methods are both synchronized

The first resource class A have the M1 method uses the second resource Class B and invokes the M2 Method on the second resource Class B to get its work done

```
public synchronized void m1(B b) {
System.out.println("M1 Method of Class A");

System.out.println("Invoking M2 Method of Class B");
b.m2();
}
```

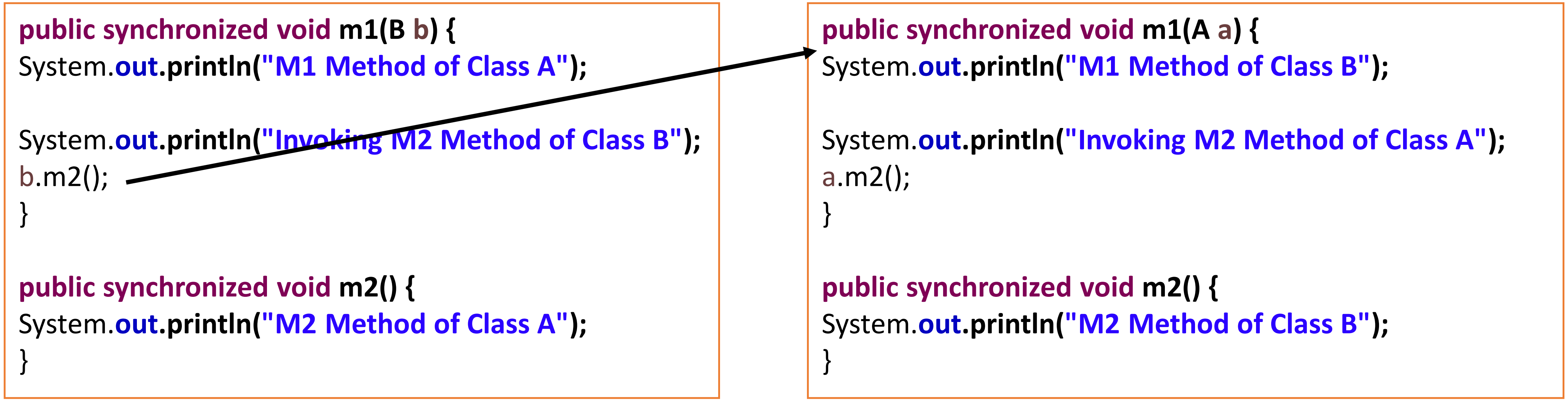
Similarly the second resource class B will use the first resource class A and invokes the M2 Method on the second resource Class A to get its work done

```
public synchronized void m1(A a) {
System.out.println("M1 Method of Class B");

System.out.println("Invoking M2 Method of Class A");
a.m2();
}
```

Suddenly, if two threads are swapped each other

The first thread enters class A M1 method then it will aquire a lock object on those two synchronized methods



```
public synchronized void m1(B b) {
System.out.println("M1 Method of Class A");

System.out.println("Invoking M2 Method of Class B");
b.m2();
}

public synchronized void m2() {
System.out.println("M2 Method of Class A");
}
```

```
public synchronized void m1(A a) {
System.out.println("M1 Method of Class B");

System.out.println("Invoking M2 Method of Class A");
a.m2();
}

public synchronized void m2() {
System.out.println("M2 Method of Class B");
}
```

But, before it reaches the point b.m2(); for acquiring the lock on the second resource class B

Then the second thread would have obtained the lock on second resource class B

Finally, they will end up waiting for each other locks to be released, which will neven happen that is dead lock

```
class A{

public synchronized void m1() {
System.out.println("M1 Method of Class A");
}

public synchronized void m2() {
System.out.println("M2 Method of Class A");
}


}
```

```
class B{

public synchronized void m1() {
System.out.println("M1 Method of Class B");
}

public synchronized void m2() {
System.out.println("M2 Method of Class B");
}


}
```

```
public class Client {

public static void main(String[] args) {

new Client();


}
}
```

```
class A{

public synchronized void m1(B b) {
System.out.println("M1 Method of Class A");

System.out.println("Invoking M2 Method of Class B");
b.m2();
}

public synchronized void m2() {
System.out.println("M2 Method of Class A");
}
}
```

```
class B{

public synchronized void m1(A a) {
System.out.println("M1 Method of Class B");

System.out.println("Invoking M2 Method of Class A");
a.m2();
}

public synchronized void m2() {
System.out.println("M2 Method of Class B");
}
}
```

```
public class Client implements Runnable {

public static void main(String[] args) {
new Client();
}

@Override
public void run() {
}
}
```

```java
class A{

public synchronized void m1(B b) {
System.out.println("M1 Method of Class A");

System.out.println("Invoking M2 Method of Class B");
b.m2();
}

public synchronized void m2() {
System.out.println("M2 Method of Class A");
}
}
```

```java
class B{

public synchronized void m1(A a) {
System.out.println("M1 Method of Class B");

System.out.println("Invoking M2 Method of Class A");
a.m2();
}

public synchronized void m2() {
System.out.println("M2 Method of Class B");
}
}
```

```java
public class Client implements Runnable {

A a = new A();
B b = new B();

public static void main(String[] args) {
new Client();
}
@Override
public void run() {
}
}
```

```
class A{

public synchronized void m1(B b) {
System.out.println("M1 Method of Class A");

System.out.println("Invoking M2 Method of Class B");
b.m2();
}

public synchronized void m2() {
System.out.println("M2 Method of Class A");
}
}
```

```
class B{

public synchronized void m1(A a) {
System.out.println("M1 Method of Class B");

System.out.println("Invoking M2 Method of Class A");
a.m2();
}

public synchronized void m2() {
System.out.println("M2 Method of Class B");
}
}
```

```
public class Client {

A a = new A();
B b = new B();

public Client() {
b.m1(a); // com.dl.deadlock.one.B.m1(A a)
}

public static void main(String[] args) {
new Client();
}
```

```
M1 Method of Class B
Invoking M2 Method of Class A
M2 Method of Class A
```

```java
class A{

public synchronized void m1(B b) {
System.out.println("M1 Method of Class A");

System.out.println("Invoking M2 Method of Class B");
b.m2();
}

public synchronized void m2() {
System.out.println("M2 Method of Class A");
}
}
```

```java
class B{

public synchronized void m1(A a) {
System.out.println("M1 Method of Class B");

System.out.println("Invoking M2 Method of Class A");
a.m2();
}

public synchronized void m2() {
System.out.println("M2 Method of Class B");
}
}
```

```java
public class Client implements Runnable {

A a = new A();
B b = new B();

public Client() {
new Thread(this).start();
b.m1(a); //com.dl.deadlock.one.B.m1(A a)
}

public static void main(String[] args) {
new Client();
}
@Override
public void run() {
a.m1(b); //com.dl.deadlock.one.A.m1(B b)
}
}
```

```
M1 Method of Class B
Invoking M2 Method of Class A
M1 Method of Class A
Invoking M2 Method of Class B
```