

06102124

URBAN
EDGE

→ Multi Threading

class BMS extends Thread

public void run()

```
for (int = i ; i <= 5 ; i++)
```

```
System.out.println ("BMS college  
engineering "+ i);
```

by
K

Thread.sleep(10000);

1

catch (InterruptedException e)

2

2. printStackTrace();

1

> class CS extends Thread

public void run()

~~for (int i=1; i<=5; i++)~~

1

```
System.out.println("CSE "+i);
```

~~toy~~

Thread.sleep(2000);

```
> catch(InterruptedException e)
```

e. printStackTrace();

class ThreadMain

2

public static void main(String args[])

BMS b1 = new BMS();

CSE c1 = new CSE();

b1.start();

c1.start();

output:

BMS - college of engineering

CSE1

CSE2

CSE3

CSE4

CSE5

BMS college of engineering

BMS college of engineering 3

BMS college of engineering 4

BMS college of engineering 5

Inter process communication

URBAN
EDGE

→ class Q {

int n;

boolean valueSet = false;

synchronized int get() {

while (!valueSet)

try {

System.out.println("In consumer waiting\n");

wait();

} catch (InterruptedException e) {

System.out.println("Interrupted exception caught");

System.out.println("Got: " + n);

valueSet = true;

System.out.println("In Intimate Producer\n");

notify();

return n;

}

Synchronized void put(int n) {

while (!valueSet)

try {

System.out.println("In Producer waiting\n");

wait();

} catch (InterruptedException e) {

System.out.println("Interrupted exception caught");

}

this.n = n;

valueSet = true;

System.out.println("Put: " + n);

System.out.println("In Intimate Consumer\n");

notify();

}

new : Consumer (v)
System.out.println ("ress Control - (to stop:)")
y
but put) —

Put : 1

Get : 1

Put : 2

Get : 2

Put : 3

Get : 3

Put : 4

Get : 4

Put : 5

Get : 5