Inter Thread Communication:

If **one Thread** wants to communicate with **other Thread** is called as **Inter Thread Communication.**

To perform **Inter Thread Communication** we have to use the following methods.
public void wait()
public void notify()
public void notifyAll()

wait()method, notify()method, notifyAll() method belongs to **java.lang.Object class not in java.lang.Thread class**

wait() method

Where **wait() method** can be used to keep a **Thread** in **waiting state** unti the **other Thread completes its work** on the same object.

Note:
The **wait() method** tells the calling **Thread** to give up the monitor and go to **sleep** until some other **Thread** enters the same monitor.

After completion of monitor then it calls the **notify() or notifyAll()** to run the waited Thread.

notify() method

Where **notify() metod** is used to notify or **wake up the Thread** that **was waiting on the same object.**

notifyAll() method

Where **notifyAll() method** can be used to give notification to **all Threads** which **are in waiting state.**

We can call **wait(), notify(), notifyAll() methods** from **Synchronized** area only, otherwise we will get **Runtime Exception** saying **IllegalMonitorStateException.**

```
public class A implements Runnable {

@Override
public void run() {

System.out.println("Run Method of Class A");

synchronized (this) {
System.out.println("Synchronized Block of Class A");

for (int i = 1; i < 5; i++) {
System.out.println(i + " Loop Inside Run Method and Synchronized Method");
}
}
}
}
```

```
Object A ref: com.dl.interthread.A@626b2d4a
Start Method
Synchronized Block of Class B
1 Loop inside Synchronized Method
2 Loop inside Synchronized Method
3 Loop inside Synchronized Method
4 Loop inside Synchronized Method
Run Method of Class A
Synchronized Block of Class A
1 Loop Inside Run Method and Synchronized Method
2 Loop Inside Run Method and Synchronized Method
3 Loop Inside Run Method and Synchronized Method
4 Loop Inside Run Method and Synchronized Method
```

```
public class B {

public static void main(String[] args) {

A a = new A();
System.out.println("Object A ref: " + a);

Thread t = new Thread(a);
t.start();
System.out.println("Start Method");

synchronized (a) {
System.out.println("Synchronized Block of Class B");

for (int i = 1; i < 5; i++) {
System.out.println(i + " Loop inside Synchronized Block");
}
}
}
}
```

```
public class A implements Runnable {

@Override
public void run() {

System.out.println("04 Run Method of Class A");

synchronized (this) {
System.out.println("05 Synchronized Block of Class A");
for (int i = 1; i < 5; i++) {
System.out.println(i + " 06  Loop Inside Run Method and Synchronized Method");
}
notify();
}
}
}
```

```
O1 bject A ref: com.dl.interthread.two.A@626b2d4a
02 Start Method
03 Synchronized Block of Class B
04 Run Method of Class A
05 Synchronized Block of Class A
1 06  Loop Inside Run Method and Synchronized Method
2 06  Loop Inside Run Method and Synchronized Method
3 06  Loop Inside Run Method and Synchronized Method
4 06  Loop Inside Run Method and Synchronized Method
1 07  Loop inside Synchronized Method
2 07  Loop inside Synchronized Method
3 07  Loop inside Synchronized Method
4 07  Loop inside Synchronized Method
```

```
public class B {

public static void main(String[] args) throws InterruptedException {

A a = new A();
System.out.println("O1 bject A ref: " + a);

Thread t = new Thread(a);
t.start();
System.out.println("02 Start Method");

synchronized (a) {
System.out.println("03 Synchronized Block of Class B");
a.wait();
for (int i = 1; i < 5; i++) {
System.out.println(i + " 07  Loop inside Synchronized Method");
}
}
}
}
```