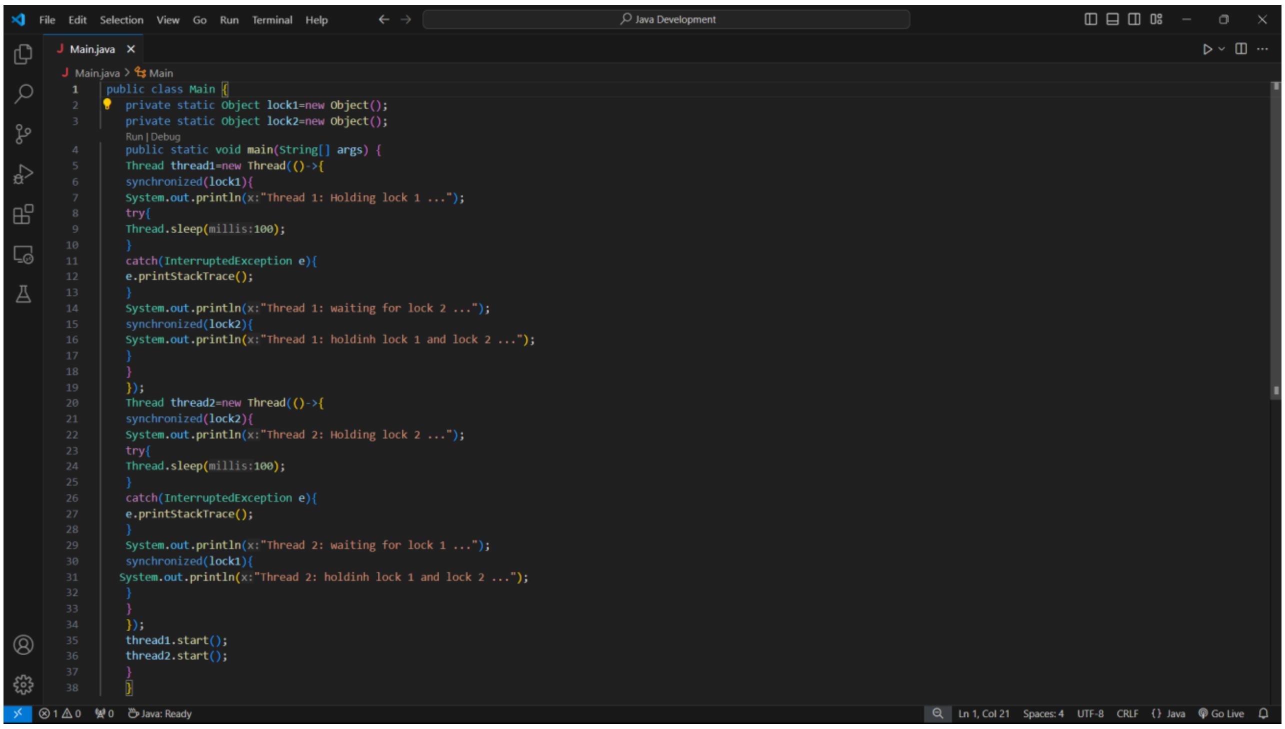
SUBMITTED BY: - **KANISHK**

2205130

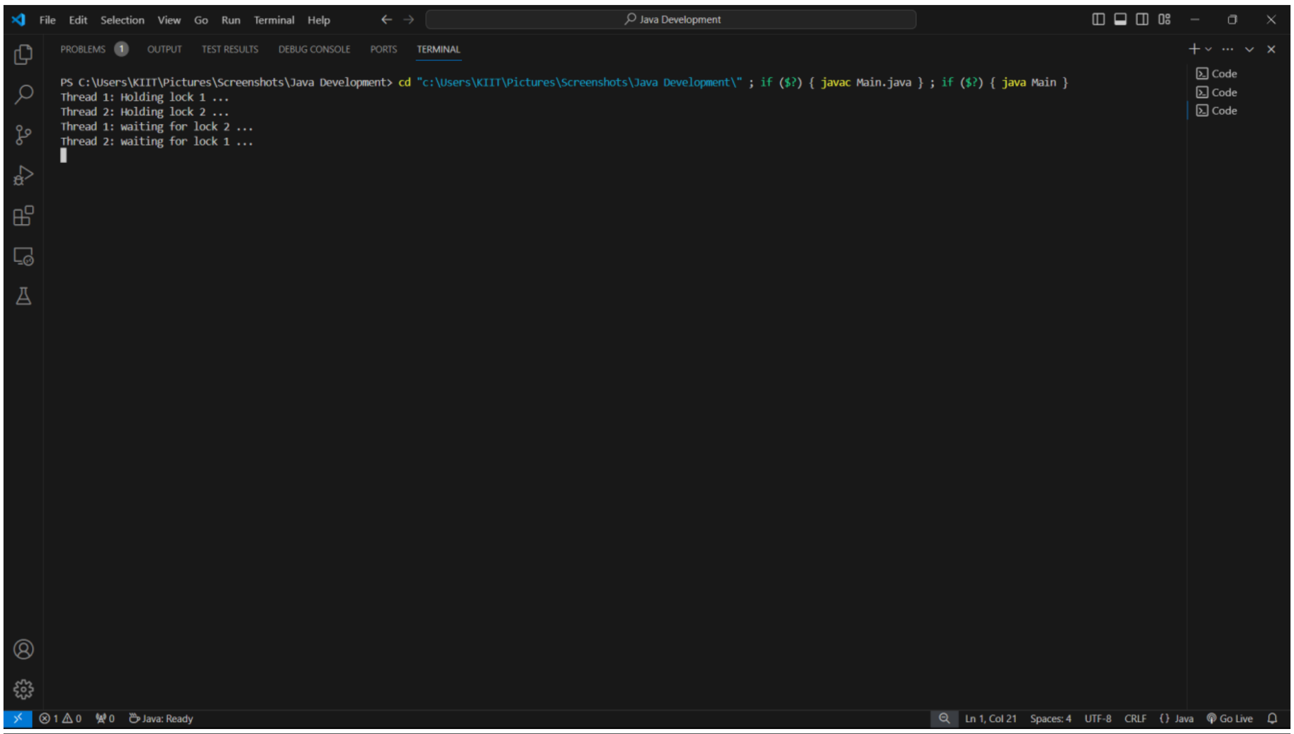
CSE-37

<u>Q1-</u> . Demonstrate Deadlock scenario in java using Simple Example



```java
public class Main {
    private static Object lock1=new Object();
    private static Object lock2=new Object();
    public static void main(String[] args) {
    Thread thread1=new Thread(()->{
    synchronized(lock1){
    System.out.println(x:"Thread 1: Holding lock 1 ...");
    try{
    Thread.sleep(millis:100);
    }
    catch(InterruptedException e){
    e.printStackTrace();
    }
    System.out.println(x:"Thread 1: waiting for lock 2 ...");
    synchronized(lock2){
    System.out.println(x:"Thread 1: holdinh lock 1 and lock 2 ...");
    }
    }
    });
    Thread thread2=new Thread(()->{
    synchronized(lock2){
    System.out.println(x:"Thread 2: Holding lock 2 ...");
    try{
    Thread.sleep(millis:100);
    }
    catch(InterruptedException e){
    e.printStackTrace();
    }
    System.out.println(x:"Thread 2: waiting for lock 1 ...");
    synchronized(lock1){
   System.out.println(x:"Thread 2: holdinh lock 1 and lock 2 ...");
    }
    }
    });
    thread1.start();
    thread2.start();
    }
}
```

<u>OUTPUT: -</u>

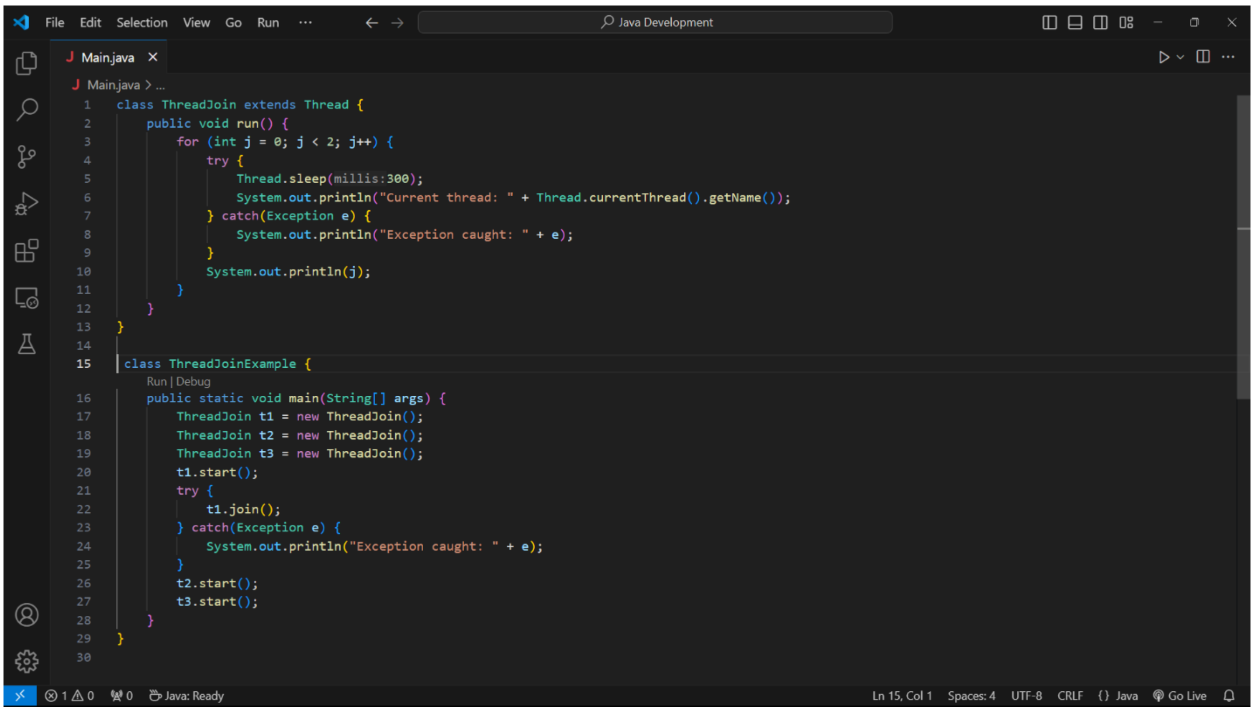


```
PS C:\Users\KIIT\Pictures\Screenshots\Java Development> cd "c:\Users\KIIT\Pictures\Screenshots\Java Development\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
Thread 1: Holding lock 1 ...
Thread 2: Holding lock 2 ...
Thread 1: waiting for lock 2 ...
Thread 2: waiting for lock 1 ...
```

Q2: - Explain Utility of Join method through an Example

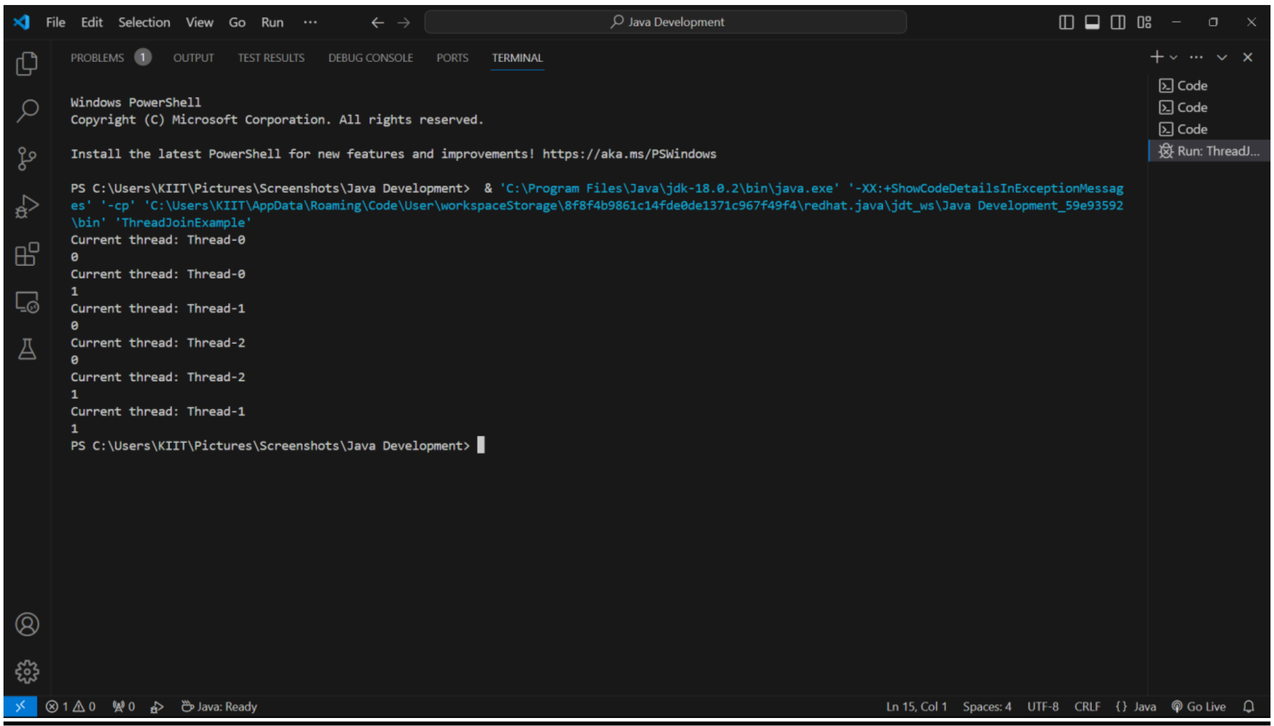
Ans: - The join() method in Java is used in two contexts:

Thread.join(): This method allows one thread to wait until another thread completes its execution. If ‘t’ is a Thread object whose thread is currently executing, then t.join() causes the current thread to pause execution until t’s thread terminates. Overloads of join() allow the programmer to specify a waiting period. Here’s an example:-



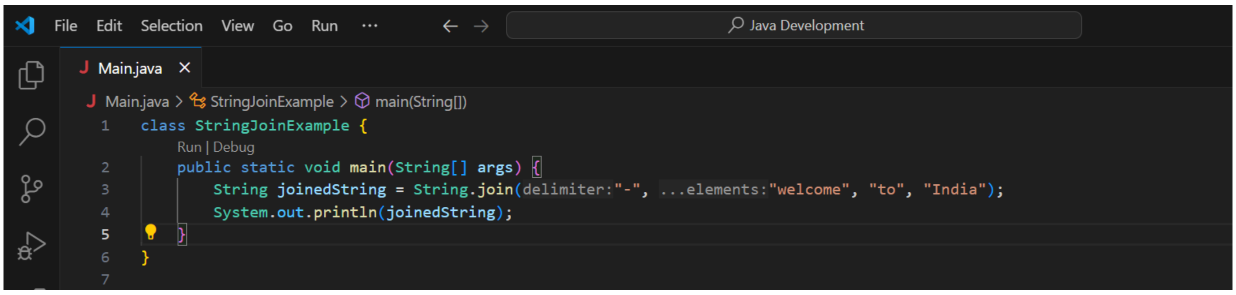
```java
class ThreadJoin extends Thread {
    public void run() {
        for (int j = 0; j < 2; j++) {
            try {
                Thread.sleep(millis:300);
                System.out.println("Current thread: " + Thread.currentThread().getName());
            } catch(Exception e) {
                System.out.println("Exception caught: " + e);
            }
            System.out.println(j);
        }
    }
}

class ThreadJoinExample {
    Run | Debug
    public static void main(String[] args) {
        ThreadJoin t1 = new ThreadJoin();
        ThreadJoin t2 = new ThreadJoin();
        ThreadJoin t3 = new ThreadJoin();
        t1.start();
        try {
            t1.join();
        } catch(Exception e) {
            System.out.println("Exception caught: " + e);
        }
        t2.start();
        t3.start();
    }
}
```

OUTPUT: -



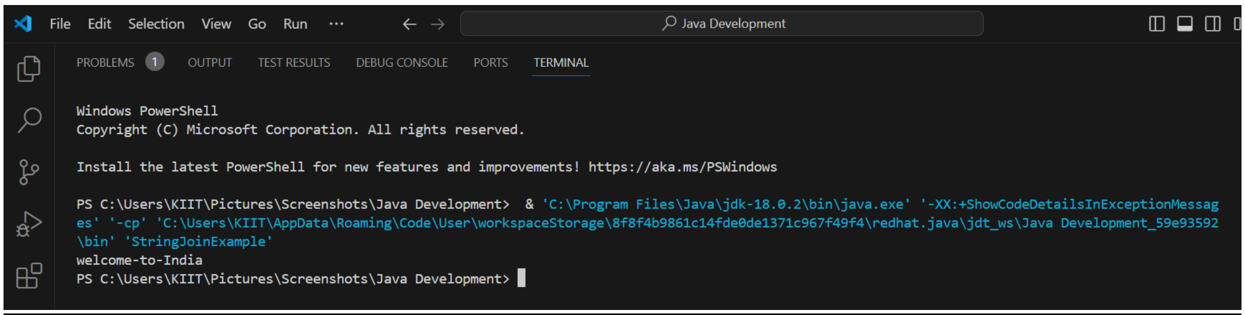
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\KIIT\Pictures\Screenshots\Java Development>  & 'C:\Program Files\Java\jdk-18.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessag
es' '-cp' 'C:\Users\KIIT\AppData\Roaming\Code\User\workspaceStorage\8f8f4b9861c14fde0de1371c967f49f4\redhat.java\jdt_ws\Java Development_59e93592
\bin' 'ThreadJoinExample'
Current thread: Thread-0
0
Current thread: Thread-0
1
Current thread: Thread-1
0
Current thread: Thread-2
0
Current thread: Thread-2
1
Current thread: Thread-1
1
PS C:\Users\KIIT\Pictures\Screenshots\Java Development>
```

<u>String.join():</u> This method concatenates the given elements with the delimiter and returns the concatenated string. If an element is null, then “null” is added. Here’s an example:



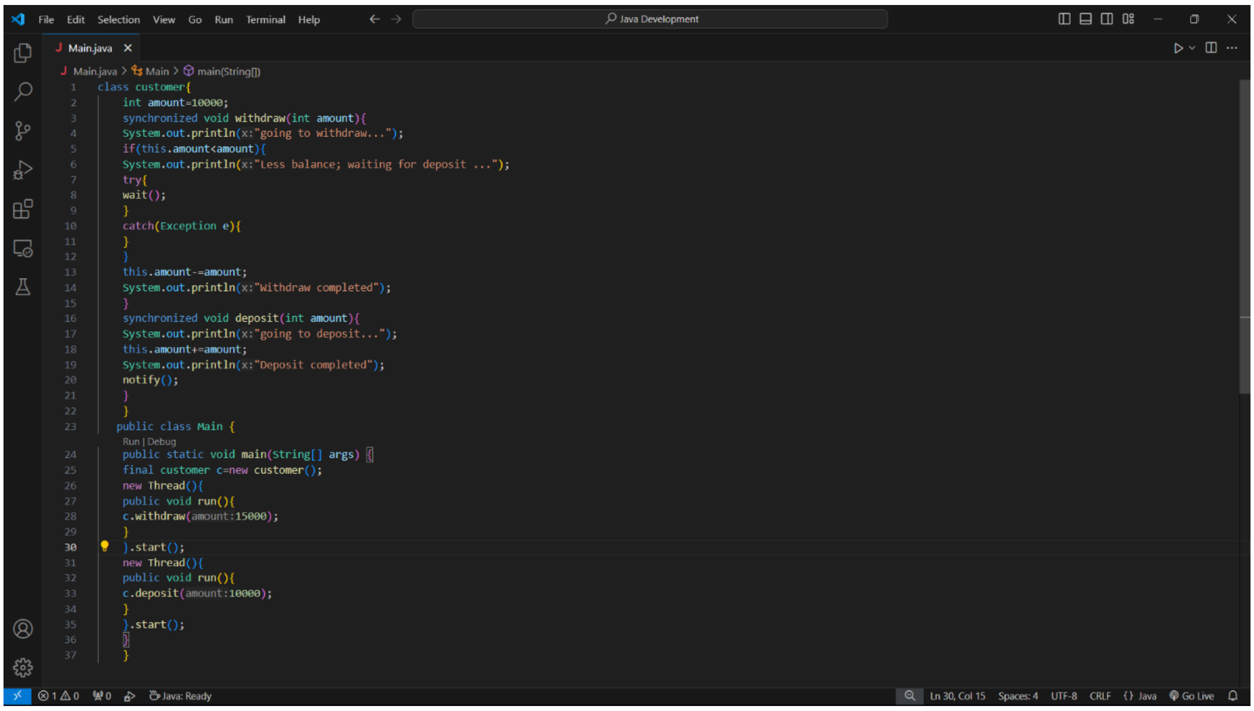
```java
class StringJoinExample {
    public static void main(String[] args) {
        String joinedString = String.join(delimiter:"-", ...elements:"welcome", "to", "India");
        System.out.println(joinedString);
    }
}
```

<u>OUTPUT: -</u>



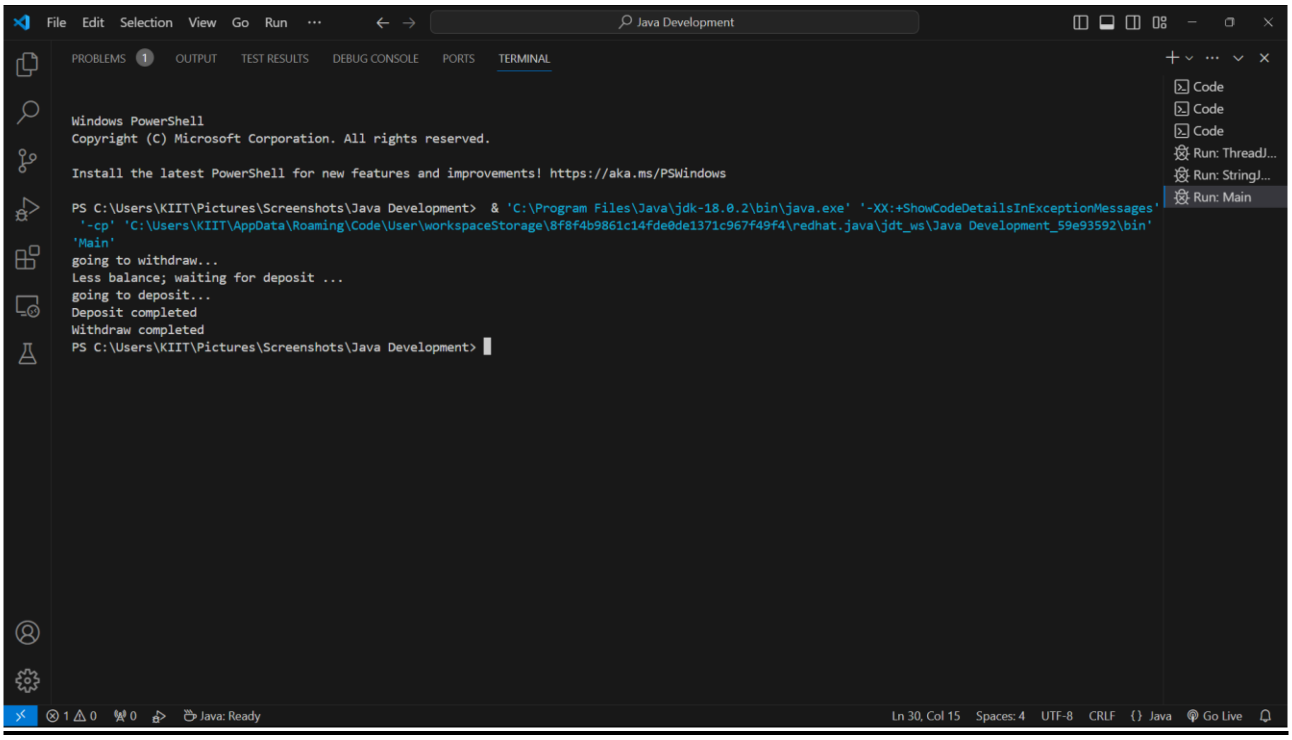
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\KIIT\Pictures\Screenshots\Java Development>  & 'C:\Program Files\Java\jdk-18.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\KIIT\AppData\Roaming\Code\User\workspaceStorage\8f8f4b9861c14fde0de1371c967f49f4\redhat.java\jdt_ws\Java Development_59e93592\bin' 'StringJoinExample'
welcome-to-India
PS C:\Users\KIIT\Pictures\Screenshots\Java Development> 
```

<u>Q3: -</u> Explain Thread Communication with Example



```java
class customer{
    int amount=10000;
    synchronized void withdraw(int amount){
    System.out.println(x:"going to withdraw...");
    if(this.amount<amount){
    System.out.println(x:"Less balance; waiting for deposit ...");
    try{
    wait();
    }
    catch(Exception e){
    }
    }
    this.amount-=amount;
    System.out.println(x:"Withdraw completed");
    }
    synchronized void deposit(int amount){
    System.out.println(x:"going to deposit...");
    this.amount+=amount;
    System.out.println(x:"Deposit completed");
    notify();
    }
    }
   public class Main {
    public static void main(String[] args) {
    final customer c=new customer();
    new Thread(){
    public void run(){
    c.withdraw(amount:15000);
    }
    }.start();
    new Thread(){
    public void run(){
    c.deposit(amount:10000);
    }
    }.start();
    }
    }
```

OUTPUT: -


File Edit Selection View Go Run ···
Java Development
PROBLEMS 1 OUTPUT TEST RESULTS DEBUG CONSOLE PORTS TERMINAL
Code
Code
Code
Run: ThreadJ...
Run: StringJ...
Run: Main
⊗ 1 ⚠ 0 0 Java: Ready
Ln 30, Col 15 Spaces: 4 UTF-8 CRLF {} Java Go Live

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\KIIT\Pictures\Screenshots\Java Development>  & 'C:\Program Files\Java\jdk-18.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages'
 '-cp' 'C:\Users\KIIT\AppData\Roaming\Code\User\workspaceStorage\8f8f4b9861c14fde0de1371c967f49f4\redhat.java\jdt_ws\Java Development_59e93592\bin'
'Main'
going to withdraw...
Less balance; waiting for deposit ...
going to deposit...
Deposit completed
Withdraw completed
PS C:\Users\KIIT\Pictures\Screenshots\Java Development>
```

In Java, threads can communicate with each other using the wait, notify and notifyall methods. These methods are used in conjunction with synchronized blocks to coordinate the execution of threads. Here's an explanation of each method with an example:

- **WAIT:** This method causes the current thread to wait until another thread invokes the notify() or notifyAll() method for the same object. It must be called from a synchronized context.
- **NOTIFY:** This method wakes up a single thread that is waiting on the object. If multiple threads are waiting, it is not specified which thread will be awakened. It must be called from a synchronized context.
- **notifyAll:** This method wakes up all threads that are waiting on the object. It must be called from a synchronized context.