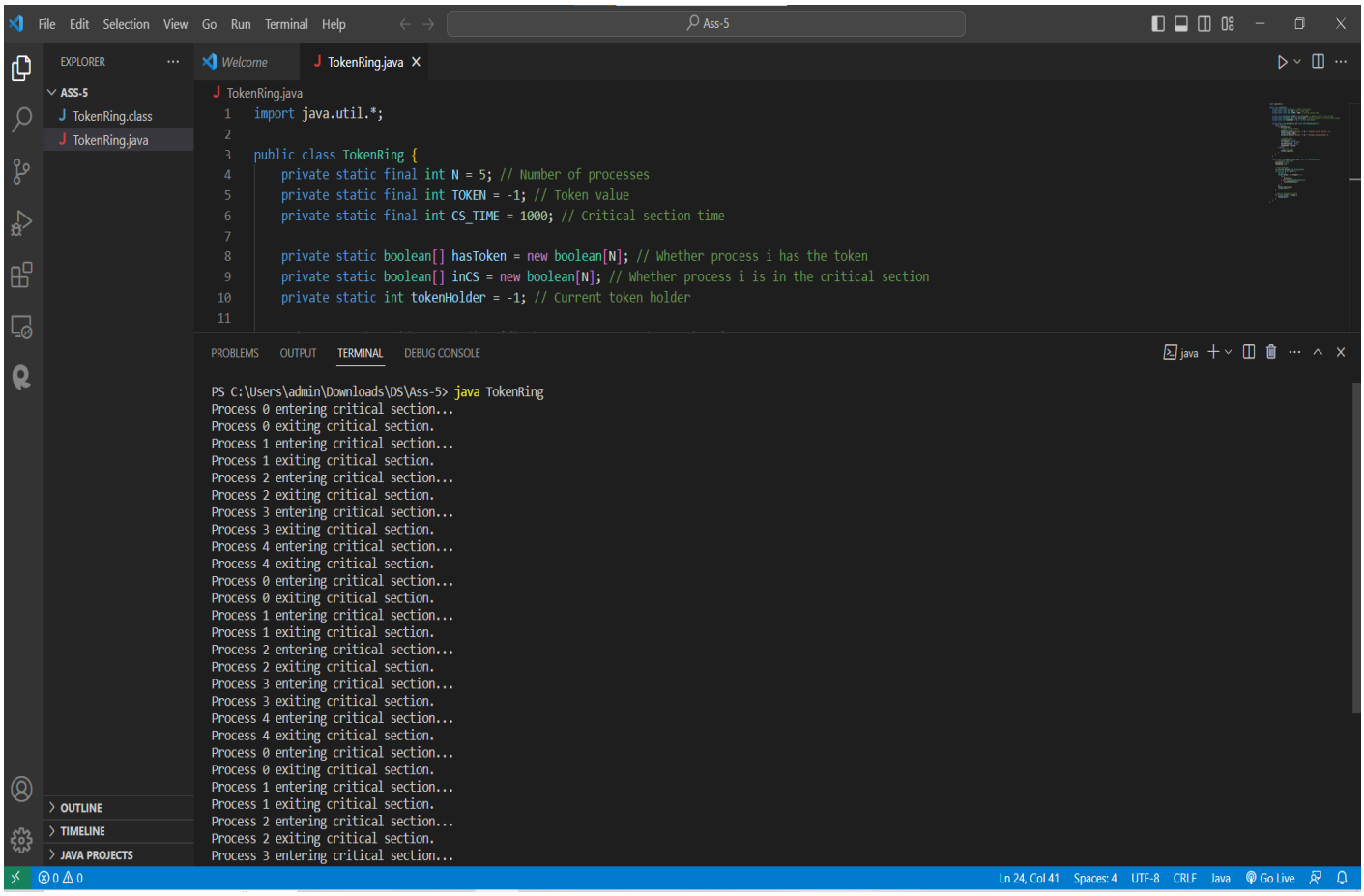
Assignment No 5

Code

```
import java.util.*;

public class TokenRing {
    private static final int N = 5; // Number of processes
    private static final int TOKEN = -1; // Token value
    private static final int CS_TIME = 1000; // Critical section time

    private static boolean[] hasToken = new boolean[N]; // Whether process i has the token
    private static boolean[] inCS = new boolean[N]; // Whether process i is in the critical section
    private static int tokenHolder = -1; // Current token holder

    private static void process(int id) throws InterruptedException {
        while (true) {
            if (hasToken[id]) {
                // Enter critical section
                inCS[id] = true;
                System.out.println("Process " + id + " entering critical section...");
                Thread.sleep(CS_TIME);
                System.out.println("Process " + id + " exiting critical section.");

                // Release token
                hasToken[id] = false;
                int nextId = (id + 1) % N;
                hasToken[nextId] = true;
                tokenHolder = nextId;
            } else {
                // Wait for token
                Thread.sleep(100);
            }
        }
    }

    public static void main(String[] args) throws InterruptedException {
        // Initialize token holder
        hasToken[0] = true;
        tokenHolder = 0;

        // Start processes
        List<Thread> threads = new ArrayList<>();
        for (int i = 0; i < N; i++) {
            int id = i;
            Thread thread = new Thread(() -> {
                try {
```

```
                process(id);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        });
        threads.add(thread);
        thread.start();
    }

    // Wait for processes to finish
    for (Thread thread : threads) {
        thread.join();
    }
  }
}
```

Output


File Edit Selection View Go Run Terminal Help
Ass-5
EXPLORER
ASS-5
TokenRing.class
TokenRing.java
Welcome
TokenRing.java
TokenRing.java
1 import java.util.*;
2
3 public class TokenRing {
4 private static final int N = 5; // Number of processes
5 private static final int TOKEN = -1; // Token value
6 private static final int CS_TIME = 1000; // Critical section time
7
8 private static boolean[] hasToken = new boolean[N]; // Whether process i has the token
9 private static boolean[] inCS = new boolean[N]; // Whether process i is in the critical section
10 private static int tokenHolder = -1; // Current token holder
11
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
java
PS C:\Users\admin\Downloads\DS\Ass-5> java TokenRing
Process 0 entering critical section...
Process 0 exiting critical section.
Process 1 entering critical section...
Process 1 exiting critical section.
Process 2 entering critical section...
Process 2 exiting critical section.
Process 3 entering critical section...
Process 3 exiting critical section.
Process 4 entering critical section...
Process 4 exiting critical section.
Process 0 entering critical section...
Process 0 exiting critical section.
Process 1 entering critical section...
Process 1 exiting critical section.
Process 2 entering critical section...
Process 2 exiting critical section.
Process 3 entering critical section...
Process 3 exiting critical section.
Process 4 entering critical section...
Process 4 exiting critical section.
Process 0 entering critical section...
Process 0 exiting critical section.
Process 1 entering critical section...
Process 1 exiting critical section.
Process 2 entering critical section...
Process 2 exiting critical section.
Process 3 entering critical section...
OUTLINE
TIMELINE
JAVA PROJECTS
0 0
Ln 24, Col 41 Spaces: 4 UTF-8 CRLF Java Go Live

Assignment No 6

Code

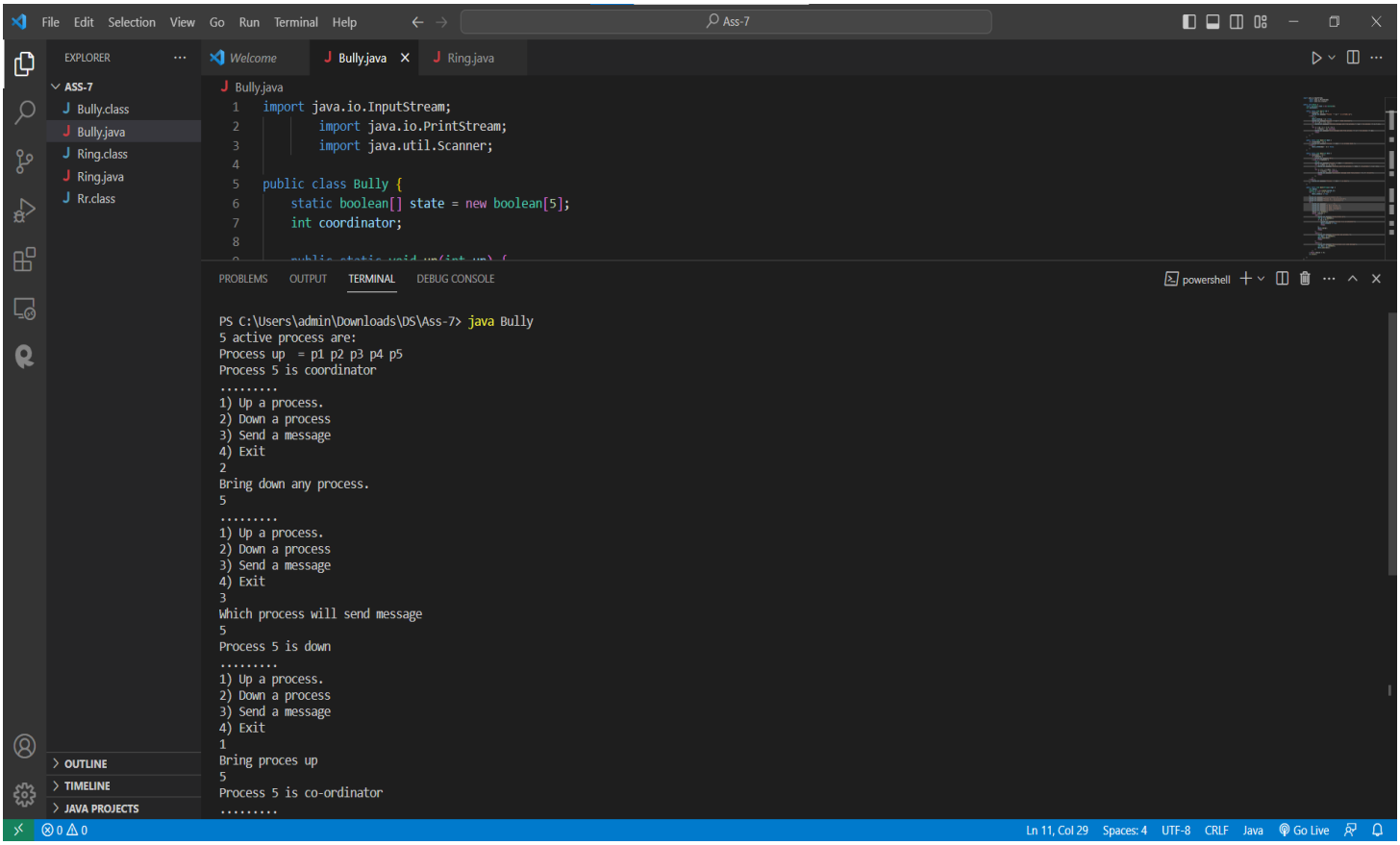
A) Bully Algorithm

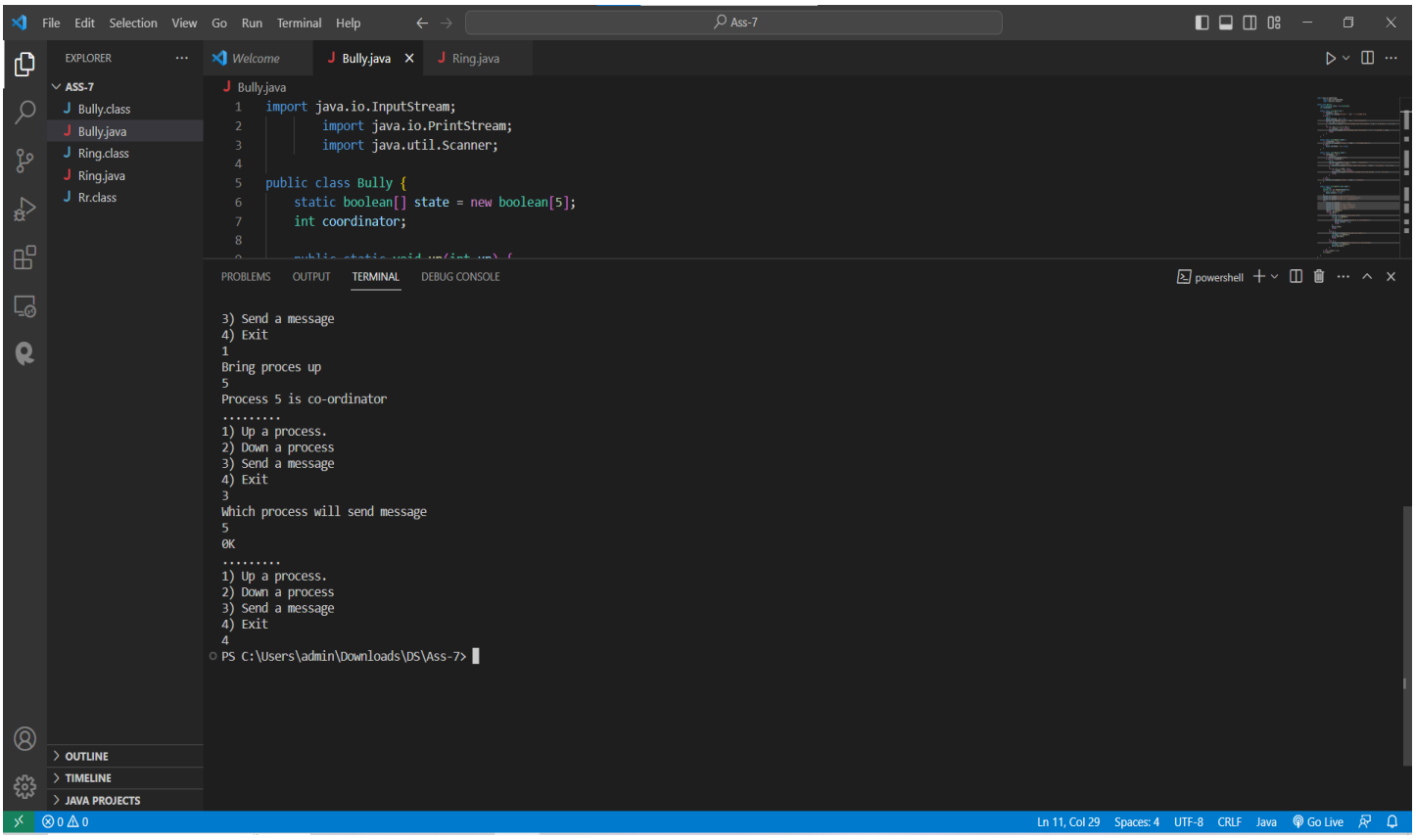
```
import java.io.InputStream;
	import java.io.PrintStream;
	import java.util.Scanner;

public class Bully {
   static boolean[] state = new boolean[5];
   int coordinator;

   public static void up(int up) {
      if (state[up - 1]) {
         System.out.println("Process " + up + " is already up");
      } else {
         int i;
         Bully.state[up - 1] = true;
         System.out.println("Process " + up + " held election");
         for (i = up; i < 5; ++i) {
            System.out.println("Election message sent from process " + up + " to process " + (i +
1));
         }
         for (i = up + 1; i <= 5; ++i) {
            if (!state[i - 1]) continue;
            System.out.println("Alive message send from process " + i + " to process " + up);
            break;
         }
      }
   }

   public static void down(int down) {
      if (!state[down - 1]) {
         System.out.println("Process " + down + " is already dowm.");
      } else {
         Bully.state[down - 1] = false;
      }
   }

   public static void mess(int mess) {
      if (state[mess - 1]) {
         if (state[4]) {
            System.out.println("0K");
         } else if (!state[4]) {
```

```
            int i;
            System.out.println("Process " + mess + " election");
            for (i = mess; i < 5; ++i) {
                System.out.println("Election send from process " + mess + " to process " + (i + 1));
            }
            for (i = 5; i >= mess; --i) {
                if (!state[i - 1]) continue;
                System.out.println("Coordinator message send from process " + i + " to all");
                break;
            }
        }
    } else {
        System.out.println("Process " + mess + " is down");
    }
}

public static void main(String[] args) {
    int choice;
    Scanner sc = new Scanner(System.in);
    for (int i = 0; i < 5; ++i) {
        Bully.state[i] = true;
    }
    System.out.println("5 active process are:");
    System.out.println("Process up  = p1 p2 p3 p4 p5");
    System.out.println("Process 5 is coordinator");
    do {
        System.out.println(".....");
        System.out.println("1) Up a process.");
        System.out.println("2) Down a process");
        System.out.println("3) Send a message");
        System.out.println("4) Exit");
        choice = sc.nextInt();
        switch (choice) {
            case 1: {
                System.out.println("Bring proces up");
                int up = sc.nextInt();
                if (up == 5) {
                    System.out.println("Process 5 is co-ordinator");
                    Bully.state[4] = true;
                    break;
                }
                Bully.up(up);
                break;
            }
            case 2: {
                System.out.println("Bring down any process.");
```

```
                int down = sc.nextInt();
                Bully.down(down);
                break;
            }
            case 3: {
                System.out.println("Which process will send message");
                int mess = sc.nextInt();
                Bully.mess(mess);
            }
        }
    } while (choice != 4);
    sc.close();

  }
}
```

Output


File Edit Selection View Go Run Terminal Help
Ass-7
EXPLORER
ASS-7
Bully.class
Bully.java
Ring.class
Ring.java
Rr.class
Welcome
Bully.java
Ring.java
import java.io.InputStream;
import java.io.PrintStream;
import java.util.Scanner;
public class Bully {
static boolean[] state = new boolean[5];
int coordinator;
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS C:\Users\admin\Downloads\DS\Ass-7> java Bully
5 active process are:
Process up = p1 p2 p3 p4 p5
Process 5 is coordinator
.....
1) Up a process.
2) Down a process
3) Send a message
4) Exit
2
Bring down any process.
5
.....
1) Up a process.
2) Down a process
3) Send a message
4) Exit
3
Which process will send message
5
Process 5 is down
.....
1) Up a process.
2) Down a process
3) Send a message
4) Exit
1
Bring proces up
5
Process 5 is co-ordinator
.....
OUTLINE
TIMELINE
JAVA PROJECTS
Ln 11, Col 29 Spaces: 4 UTF-8 CRLF Java Go Live


File Edit Selection View Go Run Terminal Help
Ass-7
EXPLORER
ASS-7
Bully.class
Bully.java
Ring.class
Ring.java
Rr.class
Welcome
Bully.java
Ring.java
import java.io.InputStream;
import java.io.PrintStream;
import java.util.Scanner;
public class Bully {
static boolean[] state = new boolean[5];
int coordinator;
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
3) Send a message
4) Exit
1
Bring proces up
5
Process 5 is co-ordinator
.....
1) Up a process.
2) Down a process
3) Send a message
4) Exit
3
Which process will send message
5
OK
.....
1) Up a process.
2) Down a process
3) Send a message
4) Exit
4
PS C:\Users\admin\Downloads\DS\Ass-7>
OUTLINE
TIMELINE
JAVA PROJECTS
Ln 11, Col 29 Spaces: 4 UTF-8 CRLF Java Go Live

B) Ring Algorithm

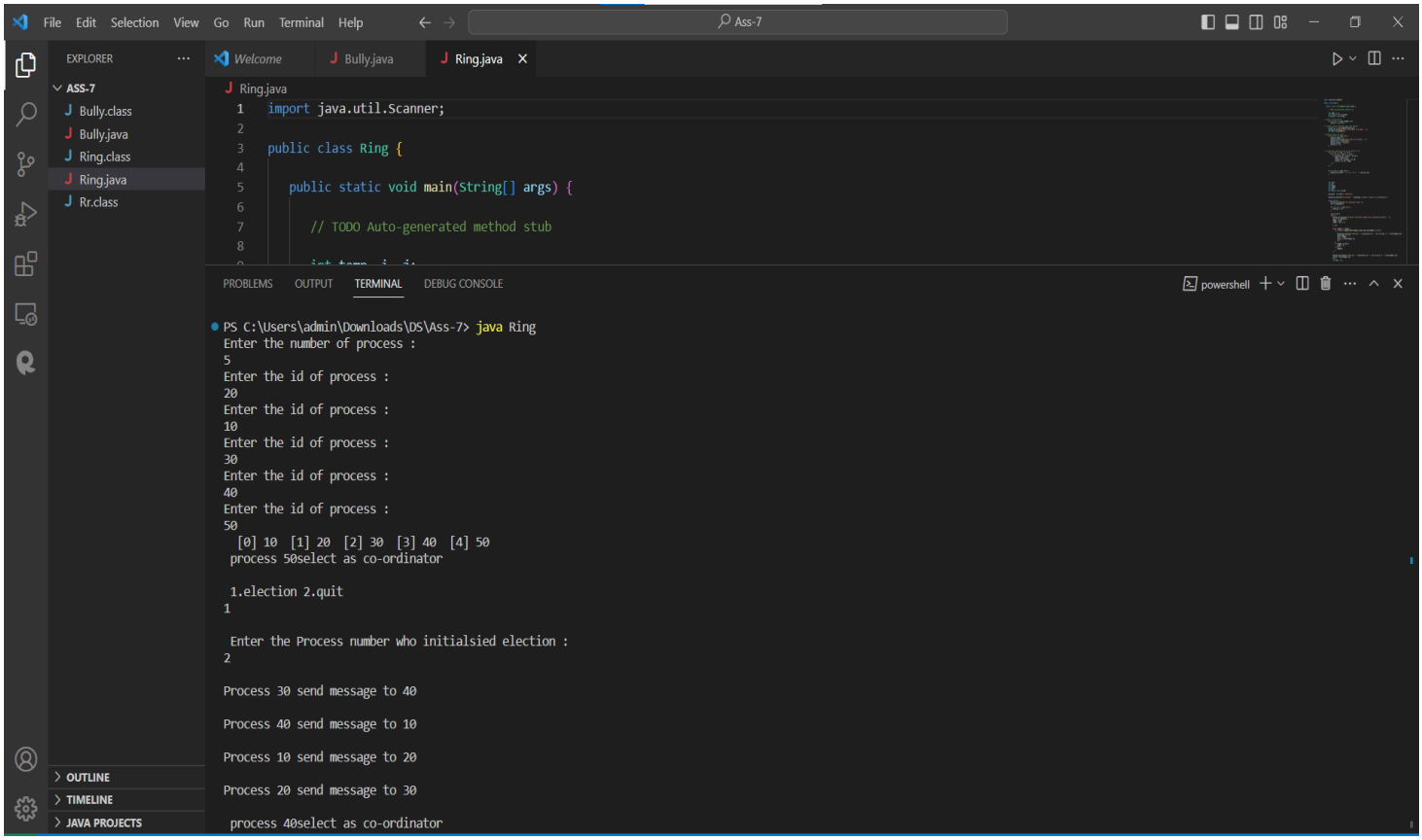
```
import java.util.Scanner;

public class Ring {

  public static void main(String[] args) {

    // TODO Auto-generated method stub

    int temp, i, j;
    char str[] = new char[10];
    Rr proc[] = new Rr[10];

// object initialisation
    for (i = 0; i < proc.length; i++)
      proc[i] = new Rr();

// scanner used for getting input from console
    Scanner in = new Scanner(System.in);
    System.out.println("Enter the number of process : ");
    int num = in.nextInt();

// getting input from users
    for (i = 0; i < num; i++) {
      proc[i].index = i;
      System.out.println("Enter the id of process : ");
      proc[i].id = in.nextInt();
      proc[i].state = "active";
      proc[i].f = 0;
    }


// sorting the processes from on the basis of id
    for (i = 0; i < num - 1; i++) {
      for (j = 0; j < num - 1; j++) {
        if (proc[j].id > proc[j + 1].id) {
          temp = proc[j].id;
          proc[j].id = proc[j + 1].id;
          proc[j + 1].id = temp;
        }
      }
    }
```

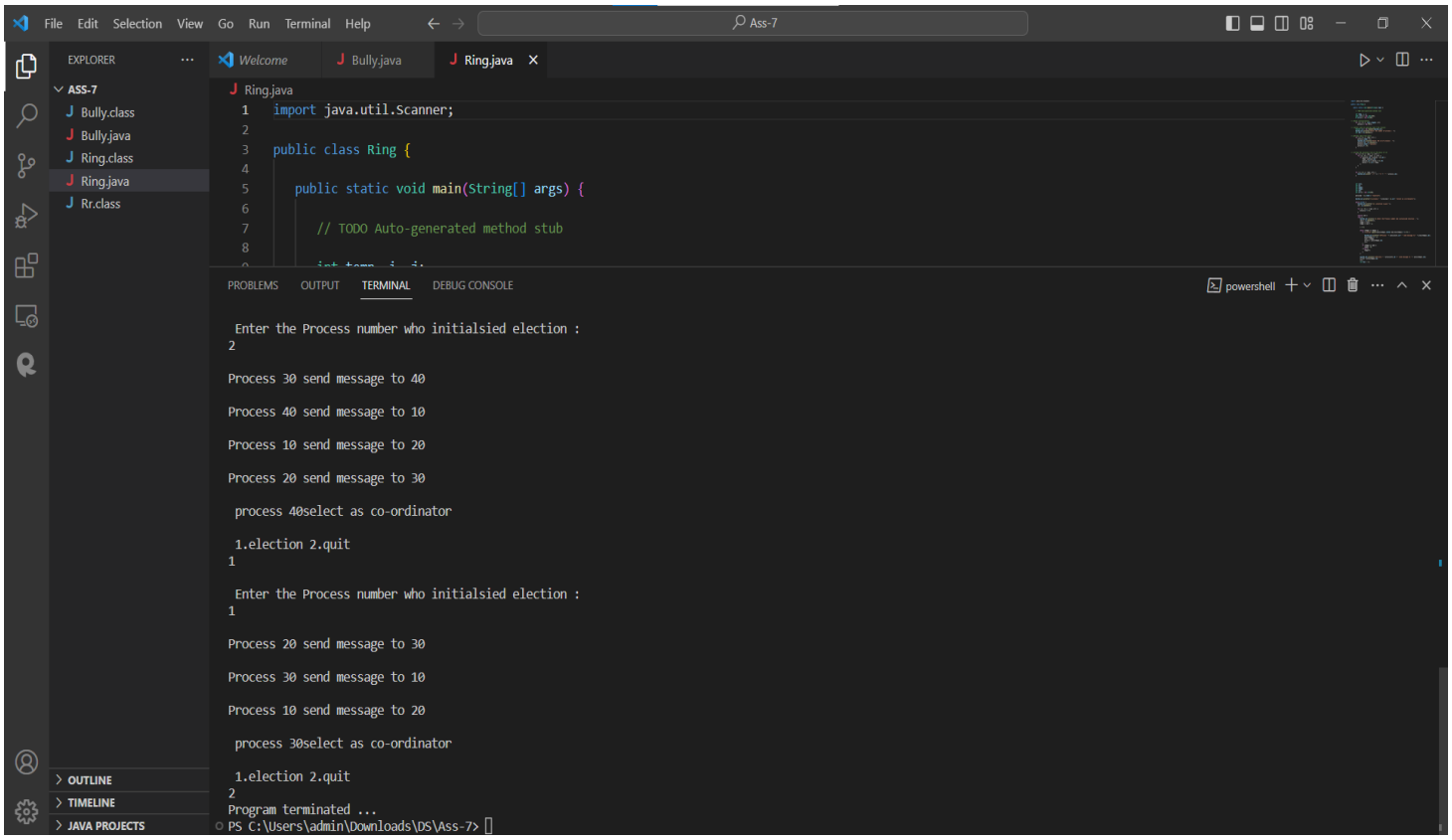
```
    for (i = 0; i < num; i++) {
      System.out.print("  [" + i + "]" + " " + proc[i].id);
    }



    int init;
    int ch;
    int temp1;
    int temp2;
    int ch1;
    int arr[] = new int[10];

    proc[num - 1].state = "inactive";

    System.out.println("\n process " + proc[num - 1].id + "select as co-ordinator");

    while (true) {
      System.out.println("\n 1.election 2.quit ");
      ch = in.nextInt();

      for (i = 0; i < num; i++) {
        proc[i].f = 0;
      }

      switch (ch) {
      case 1:
        System.out.println("\n Enter the Process number who initialsied election : ");
        init = in.nextInt();
        temp2 = init;
        temp1 = init + 1;

        i = 0;

        while (temp2 != temp1) {
          if ("active".equals(proc[temp1].state) && proc[temp1].f == 0) {

            System.out.println("\nProcess " + proc[init].id + " send message to " +
proc[temp1].id);
            proc[temp1].f = 1;
            init = temp1;
            arr[i] = proc[temp1].id;
            i++;
          }
          if (temp1 == num) {
```

```
            temp1 = 0;
          } else {
            temp1++;
          }
        }

        System.out.println("\nProcess " + proc[init].id + " send message to " +
proc[temp1].id);
        arr[i] = proc[temp1].id;
        i++;
        int max = -1;


// finding maximum for co-ordinator selection
        for (j = 0; j < i; j++) {
          if (max < arr[j]) {
            max = arr[j];
          }
        }

// co-ordinator is found then printing on console
        System.out.println("\n process " + max + "select as co-ordinator");


        for (i = 0; i < num; i++) {

          if (proc[i].id == max) {
            proc[i].state = "inactive";
          }
        }
        break;
      case 2:
        System.out.println("Program terminated ...");
        return ;
      default:
        System.out.println("\n invalid response \n");
        break;
      }


    }
  }

}

class Rr {
```

```
  public int index;   // to store the index of process
  public int id;      // to store id/name of process
  public int f;
  String state;       // indiactes whether active or inactive state of node

}
```

Output


File Edit Selection View Go Run Terminal Help
Ass-7
EXPLORER
ASS-7
Bully.class
Bully.java
Ring.class
Ring.java
Rr.class
Welcome
Bully.java
Ring.java
Ring.java
1 import java.util.Scanner;
3 public class Ring {
5 public static void main(String[] args) {
7 // TODO Auto-generated method stub
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS C:\Users\admin\Downloads\DS\Ass-7> java Ring
Enter the number of process :
5
Enter the id of process :
20
Enter the id of process :
10
Enter the id of process :
30
Enter the id of process :
40
Enter the id of process :
50
[0] 10 [1] 20 [2] 30 [3] 40 [4] 50
process 50select as co-ordinator
1.election 2.quit
1
Enter the Process number who initialsied election :
2
Process 30 send message to 40
Process 40 send message to 10
Process 10 send message to 20
Process 20 send message to 30
process 40select as co-ordinator
OUTLINE
TIMELINE
JAVA PROJECTS


File Edit Selection View Go Run Terminal Help
Ass-7
EXPLORER
ASS-7
Bully.class
Bully.java
Ring.class
Ring.java
Rr.class
Welcome
Bully.java
Ring.java
Ring.java
1 import java.util.Scanner;
3 public class Ring {
5 public static void main(String[] args) {
7 // TODO Auto-generated method stub
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
Enter the Process number who initialsied election :
2
Process 30 send message to 40
Process 40 send message to 10
Process 10 send message to 20
Process 20 send message to 30
process 40select as co-ordinator
1.election 2.quit
1
Enter the Process number who initialsied election :
1
Process 20 send message to 30
Process 30 send message to 10
Process 10 send message to 20
process 30select as co-ordinator
1.election 2.quit
2
Program terminated ...
PS C:\Users\admin\Downloads\DS\Ass-7>
OUTLINE
TIMELINE
JAVA PROJECTS