**ASSIGNMENT**

**-APARNA S**

**Assignment #38**

# I am Aparna s

# Team and Player With Queue

Write a multi-threaded Java program to implement Team and Player. Span two thread for team and player from the main method. The team inserts number starting from 0 to N into the queue, the consumer should take these number from the shared queue and prints the number to the console.   
The first input is the number of elements inserted into the queue.

Create a main class "Main.java"

Create **TeamThread** class extending Thread class with below attributes,

* threadMain - Main class object
* numberOfPlayers

Create a constructor for TeamThread with arguments threadMain and numberOfPlayers

Create **PlayerThread** class extending Thread class with below attributes,

* threadMain - Main class object
* numberOfPlayers

Create a constructor for PlayerThread with arguments threadMain and numberOfPlayers   
  
In the run() method the main thread object is synchronized and using the wait() and notifyAll() method   
  
Input and Output Format:   
First input corresponds to the number of players.   
Refer sample input and output for formatting specifications.   
  
**[All text in bold corresponds to input and the rest corresponds to output]**   
**Sample Input/Output :**   
Enter the number of Players   
**10**   
Player Thread 0   
Player Thread 1   
Player Thread 2   
Player Thread 3   
Player Thread 4   
Player Thread 5   
Player Thread 6   
Player Thread 7

**SOURCE-CODE**

**package** com.cg;

**import** java.util.\*;

**import** java.io.\*;

**import** java.util.PriorityQueue;

**class** TeamThread **extends** Thread

{

Thread threadMain;

**int** numberOfPlayers;

**public** TeamThread(Thread threadMain, **int** numberOfPlayers)

{

**super**();

**this**.threadMain = threadMain;

**this**.numberOfPlayers = numberOfPlayers;

}

**public** **synchronized** **void** run()

{

**for** (**int** i = 0; i < numberOfPlayers; i++)

{

Main.*queue*.add(i);

}

Main.*value* = **true**;

**try**

{

wait();

} **catch** (InterruptedException e)

{

e.printStackTrace();

}

}

}

**class** PlayerThread **extends** Thread

{

Thread threadMain;

**int** numberOfPlayers;

**public** PlayerThread(Thread threadMain, **int** numberOfPlayers)

{

**super**();

**this**.threadMain = threadMain;

**this**.numberOfPlayers = numberOfPlayers;

}

**public** **synchronized** **void** run()

{

**for** (**int** i : Main.*queue*)

{

System.***out***.println("Player thread " + i);

}

notifyAll();

}

}

**class** Main

{

**public** **static** PriorityQueue<Integer> *queue* = **new** PriorityQueue<>();

**public** **static** **boolean** *value* = **false**;

**public** **static** **void** main(String[] args)

{

System.***out***.println("Enter the number of Players");

Scanner sc = **new** Scanner(System.***in***);

**int** x = sc.nextInt();

sc.close();

Thread thread = **new** Thread();

TeamThread t\_thread = **new** TeamThread(thread, x);

t\_thread.start();

PlayerThread p\_thread = **new** PlayerThread(thread, x);

p\_thread.start();

}

}

**OUTPUT**

Enter the number of Players

10

Player thread 0

Player thread 1

Player thread 2

Player thread 3

Player thread 4

Player thread 5

Player thread 6

Player thread 7

Player thread 8

Player thread 9