Practical 1

AIM: Demonstrate round robin scheduling using threads.

CODE:

import java.io.\*;

class job implements Runnable {

int process\_id, no\_of\_instr, time\_quantum;

Thread t;

job(int pid, int instr, int tq) {

process\_id = pid;

no\_of\_instr = instr;

time\_quantum = tq;

t = new Thread(this);

t.start();

}

public void run()

{

try

{

for(int i=1;i<=no\_of\_instr;i++)

{

System.out.println("executing instr no"+i+"of process"+process\_id);

Thread.sleep(time\_quantum);

}

System.out.println("job"+process\_id+"is over");

}

catch(InterruptedException e)

{

System.out.println("the job has been interrupted....");

}

}

}

class os {

public static void main(String args[]) {

try {

int process\_id = 100, time\_quantum = 100;

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter a user process starting number:");

process\_id = Integer.parseInt(br.readLine());

System.out.println("Enter a time quantum(in millis):");

time\_quantum = Integer.parseInt(br.readLine());

job j1 = new job(++process\_id, 10, time\_quantum);

job j2 = new job(++process\_id, 6, time\_quantum);

job j3 = new job(++process\_id, 8, time\_quantum);

} catch (Exception e) {

System.out.println("Some process failed to complete...");

System.out.println("Plz contact system admin...");

}

}

}

OUTPUT:

