Article on Java Multithread

**Multithreading** is a process of executing multiple threads simultaneously. A multithreaded program contains two or more parts that can run currently. Each part of such a program is called thread. A thread is an independent path of execution within a program. Thus, multithreading is a specialized form of multitasking.

**Advantages of Multithreaded:**

1.Multithread used in games and animation.

2.It is better than multiprocessing.

3.Using multithreading ,you can perform many operation at a time.

4.It does not block the user so its saves time.

5.Threads are independent so it does not affect another threads.

6.multithread enables you to write in away when multiple activity can proceed in

the same program.

**Thread Priorities:**

1.A thread can voluntarily relinquish control.

2.A thread can be preempted by a higher-priority thread.

**Creating a thread:**

1.You can implement the Runnable interface.

2.you can extend the Thread class.

**Using Multithreading:**

1.With the careful use of multithreading, you can create efficient programs.

2. If you create too many threads ,you can actually degrade the performance of

your program rather than enhance it.

**Example on Thread:**

package oopbthread1;

public class OOPBThread1 {

public OOPBThread1() {

new MyClass();

m.start();

m.join();

MyClass3 m3=new MyClass3();

Thread t=new Thread(m3);

t.start();

for (int i = 0; i > -1; i++) {

System.out.println("Main Class");

try {

Thread.sleep(10);

} catch (Exception ex) {

};

}

System.out.println("Completed");

}

public static void main(String[] args) {

new OOPBThread1();

}

}

class MyClass3 implements Runnable {

public void run() {

while (true) {

System.out.println("This Is MyClass3");

try {

Thread.sleep(10);

} catch (Exception ex) {

};

}

}

}

class MyClass extends Thread {

public void run() {

while (true) {

System.out.println("This Is MyClass");

try {

Thread.sleep(10);

} catch (Exception ex) {

};

} }

}