Familiarizing Thread class methods:

```
class CurrentThreadDemo
{
public static void main(String args[])
{
Thread t=Thread.currentThread();
System.out.println("current thread:"+t);
t.setName("mythread");
System.out.println("After name change:"+t);
try
{
for(int n=5;n>0;n--)
{
System.out.println(n);
Thread.sleep(1000);
}
}
catch(InterruptedException e)
{
System.out.println("main thread interupted");
}
}
}
```

Creating threads:

Method 1:Implementing Runnable interface

```
class NewThread implements Runnable {
Thread t;
NewThread() {//creates a new thread
t=new Thread(this,"DemoThread");
System.out.println("child thread:"+t);
t.start(); }//starts the thread
//this is the entry point for the new thread
public void run() {
try {
for(int i=5;i>0;i--) {
System.out.println("child thread:"+i);
Thread.sleep(500); }
}
catch(InterruptedException e) {
System.out.println("child interupted"); }
System.out.println("exiting child thread"); }
}
class ThreadDemo {
public static void main(String args[]) {
new NewThread();//create a new thread
try {
for(int n=5;n>0;n--) {
System.out.println("main thread:"+n);
Thread.sleep(1000); }
}
catch(InterruptedException e) {
System.out.println("main thread interupted"); }
System.out.println("exiting main thread"); }
}
```

Method 2:Extending Thread class

```
class NewThread extends Thread {
NewThread() { //creates a new thread
super("DemoThread");
System.out.println("child thread:"+this);
start(); } //starts the thread
//this is the entry point for the new thread
public void run() {
try {
```

```
for(int i=5;i>0;i--) {
System.out.println("child thread:"+i);
Thread.sleep(500); }
}
catch(InterruptedException e) {
System.out.println("child interupted"); }
System.out.println("exiting child thread"); }
}
class ExtendThread {
public static void main(String args[]) {
new NewThread();//create a new thread
try {
for(int n=5;n>0;n--) {
System.out.println("main thread:"+n);
Thread.sleep(1000); }
}
catch(InterruptedException e) {
System.out.println("main thread interupted"); }
System.out.println("exiting main thread"); }
}
```

Multithreading implementation:

Using Runnable interface

```
class NewThread implements Runnable
{
Thread t;
String name;
NewThread(String threadname)//creates a new thread
{
name=threadname;
t=new Thread(this,name);
System.out.println("child thread:"+t);
t.start();//starts the thread
```

```
}
//this is the entry point for the new thread
public void run()
{
try
{
for(int i=5;i>0;i--)
{
System.out.println(name+" thread:"+i);
Thread.sleep(1000);
}
}
catch(InterruptedException e)
{
System.out.println(name+" interupted");
}
System.out.println(name+"exiting");
}
}
class MultiThreadDemo2
{
public static void main(String args[])
{
new NewThread("one");//create a new thread
new NewThread("two");
new NewThread("three");
try
{
Thread.sleep(10000);

}
catch(InterruptedException e)
{
```

```
System.out.println("main thread interupted");
}
System.out.println("exiting main thread");
}
}
```

Extending Thread class:

```
class NewThread extends Thread
{
String name;
NewThread(String threadname)//creates a new thread
{
super(threadname);
name=threadname;
System.out.println("child thread:"+this);
start();//starts the thread
}
//this is the entry point for the new thread
public void run()
{
try
{
for(int i=5;i>0;i--)
{
System.out.println(this.name+":"+i);
Thread.sleep(1000);
}
}
catch(InterruptedException e)
{
System.out.println(this.name+"interupted");
}
System.out.println(this.name+"exiting");
}
}
class MultiThreadDemo
{
public static void main(String args[])
{
new NewThread("one");//create first thread
new NewThread("two");//create second thread
new NewThread("three");//create third thread
try
{
Thread.sleep(10000);}
catch(InterruptedException e)
{
System.out.println("main thread interupted");
}
System.out.println("exiting main thread");
}
}
```

Thread Synchronization:

Program without synchronization:

```
class callme
{
void call(String msg)
{
System.out.print("["+msg);
try
{
Thread.sleep(1000);
}
catch(InterruptedException e)
{
```

```
System.out.println("interupted");
}
System.out.println("]");
}
}
class caller implements Runnable
{
String msg;
callme target;
Thread t;
public caller(callme targ,String s)
{
target=targ;
msg=s;
t=new Thread(this);
//System.out.println(t);
t.start();
}
public void run()
{
target.call(msg);
}
}
class nosynch
{
public static void main(String args[])
{
callme target=new callme();
caller ob1=new caller(target,"Hello");
caller ob2=new caller(target,"Synchronized");
caller ob3=new caller(target,"world");
//waits for threads to terminate
try
```

```
{
ob1.t.join();
System.out.println("finished ob1");
ob2.t.join();
System.out.println("finished ob2");
ob3.t.join();
System.out.println("finished ob3");
}
catch(InterruptedException e)
{
System.out.println("interupted");
}
}
}
```

Synchronization achieved with synchronized keyword:

```
class callme
{
synchronized void call(String msg)
{
System.out.print("["+msg);
try
{
Thread.sleep(1000);
}
catch(InterruptedException e)
{
System.out.println("interupted");
}
System.out.println("]");
}
}
class caller implements Runnable
```

```
{
String msg;
callme target;
Thread t;
public caller(callme targ,String s)
{
target=targ;
msg=s;
t=new Thread(this);
t.start();
}
public void run()
{
target.call(msg);
}
}
class synch
{
public static void main(String args[])
{
callme target=new callme();
caller ob1=new caller(target,"Hello");
caller ob2=new caller(target,"Synchronized");
caller ob3=new caller(target,"world");
//waits for threads to end
try
{
ob1.t.join();
//System.out.println("finished ob1");
ob2.t.join();
//System.out.println("finished ob2");
ob3.t.join();
//System.out.println("finished ob3");
}catch(InterruptedException e)
{
System.out.println("interupted");
}}}
```

Synchronization achieved with synchronized method:

```
class callme
{
void call(String msg)
{
System.out.print("["+msg);
try
{
Thread.sleep(1000);
}
catch(InterruptedException e)
{
System.out.println("interupted");
}
System.out.println("]");
}
}
class caller implements Runnable
{
String msg;
callme target;
Thread t;
public caller(callme targ,String s)
{
target=targ;
msg=s;
t=new Thread(this);
t.start();
}
public void run()
{
synchronized(target)
{
target.call(msg);
}
}
}
class synch2
{
public static void main(String args[])
{
callme target=new callme();
caller ob1=new caller(target,"Hello");
caller ob2=new caller(target,"Synchronized");
caller ob3=new caller(target,"world");
//waits for threads to end
try
{
ob1.t.join();
//System.out.println("finished ob1");
ob2.t.join();
//System.out.println("finished ob2");
ob3.t.join();
//System.out.println("finished ob3");
}catch(InterruptedException e)
{
System.out.println("interupted");
}
}
}
```