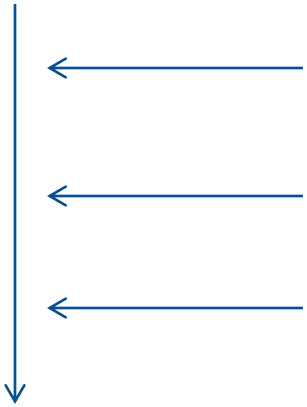
II YEAR – IV SEMESTER
COURSE CODE: 4BIT4C1
CORE COURSE VII – JAVA PROGRAMMING
UNIT 4

Multithreaded Programming

Multithreading is a conceptual programming paradigm where a program(process) is divided into two or more sub programs(Processes), which can be implemented at the same time in parallel.For Example , one subprogram can display an animation on the screen while another may build the next animation to be displayed.This is something similar to dividing a task into sub tasks and assigning them to different people for execution independently and simultaneously.

A thread is similar to a program that has a single flow of control. It has beginning ,a body and an end, and executes commands sequentially.

```
class abc
{
----------------------
----------------------
----------------------
----------------------
----------------------
}
```

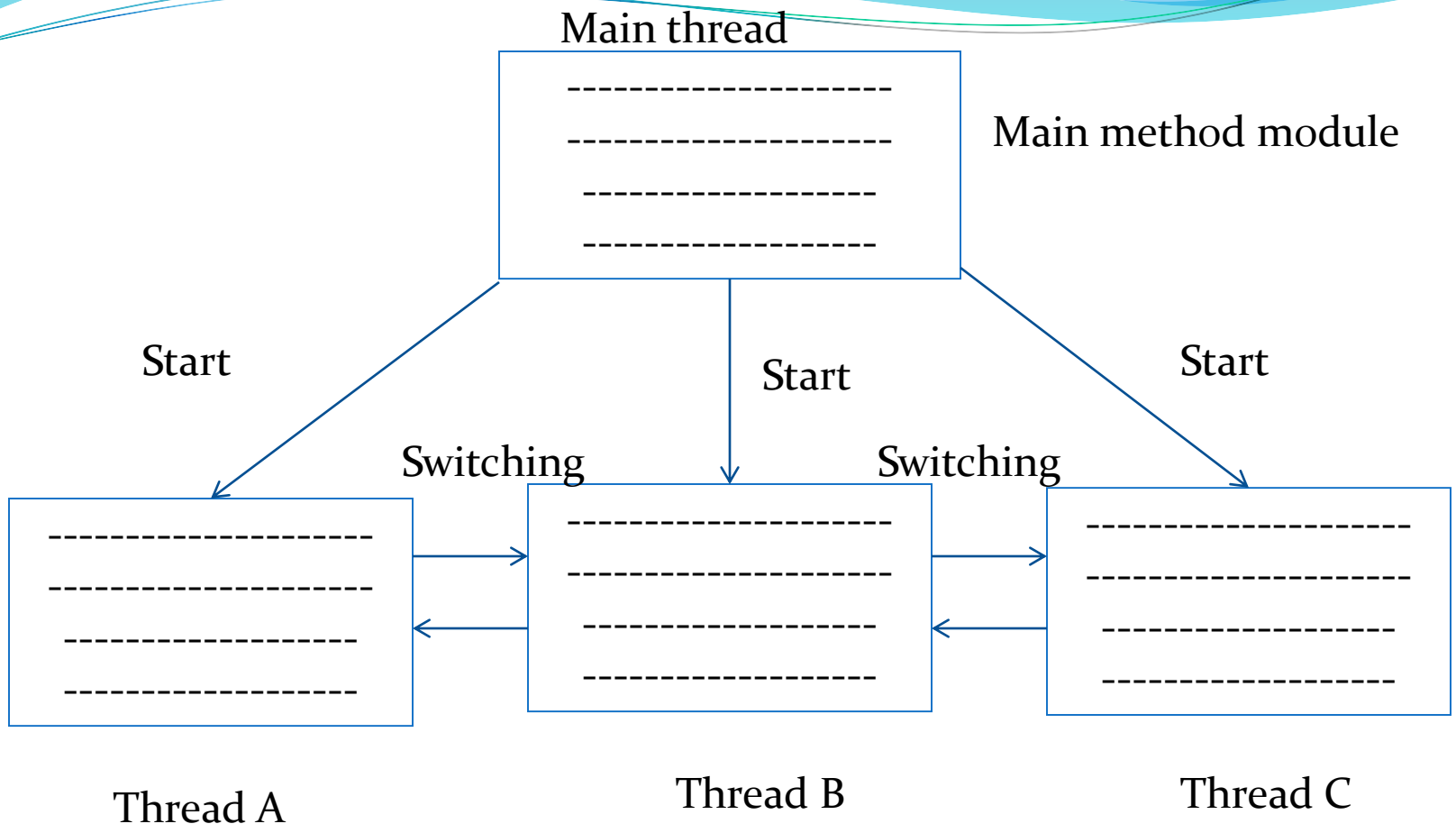


Beginning

Single –Threaded body of execution

End

Single –Threaded Program



A Multithreaded Program

Creating Threads

Threads are implemented in the form of objects that contain a method called **run().**The **run()** method is the heart and soul of any thread.It makes up the entire body of a thread and is the only method in which the threads behaviour can be implemented.

Syntax:

```
public void run()
{
---
--------- (statements for implementing thread)
---------
}
```

The **run()** method should be invoked by an object of the concerned thread. This can be achieved by creating the thread and initiating it with the help of another thread method called **start().**

A new thread can be created in two ways.

1.By creating a thread class : Define a class that extends Thread class and overrode its run() method.

2.By converting a class to a thread: Define a class that implements Runnable interface.

Declaring the class

The thread class can be extend as follows

```
class mythread extends Thread
{
.....
.....
}
```

Starting New thread

```
mythread a=new mythread();
a.start();
```

Or

```
new  mythread().start( );
```

Stopping a Thread

```
mythread.stop();
```

Blocking a Thread

A thread can also be temporarily suspended or blocked by using the following method.

sleep() : blocked for a specified time.
suspend(): blocked until further orders.
wait(): blocked until certain condition occurs.

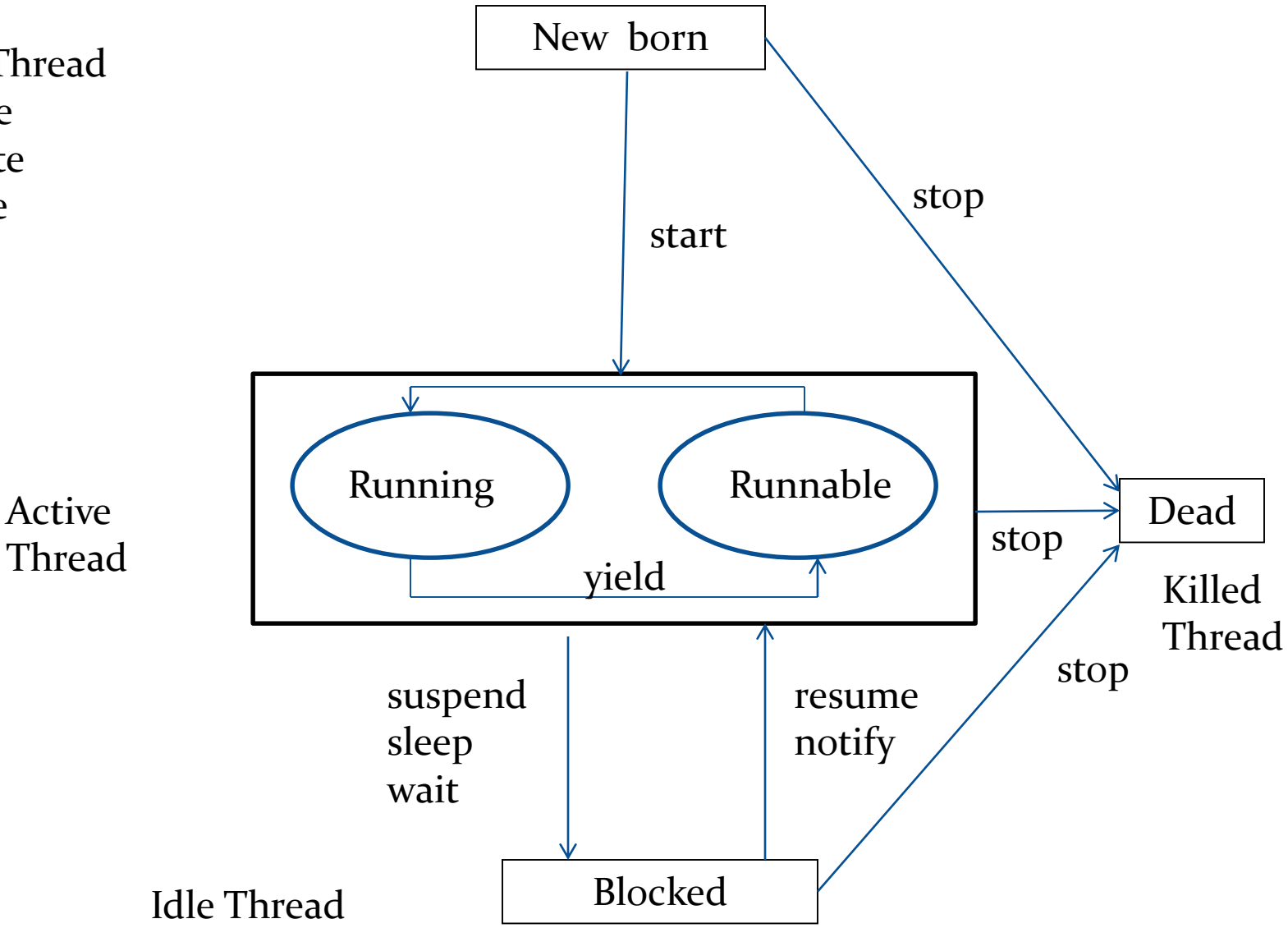
These methods cause the thread to go into the blocked state.

The thread will return to the runnable state when the specified time is elapsed in the case of sleep()

The resume method is invoked in the case of suspend()

The notify() method is called in the case of wait()

Life Cycle of a Thread
1.Newborn state
2.Runnable state
3.Running state
4.Blocked state
5.Dead state



Life Cycle of a Thread

1.Newborn state

When we create a thread object , the thread is born and is said to be in newborn state.

2.Runnable state

It means that the thread is ready for execution and is waiting for the availability of the processor.

3.Running state

the processor has given its time to the thread for its execution.

4.Blocked state

A thread is said to be blocked when it is prevented from entering into the runnable state and subsequently the running state.

5.Dead state

A running thread ends its life when it has completed executing its run() method.