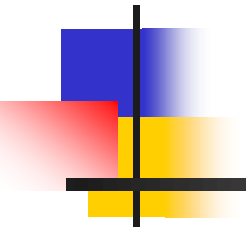
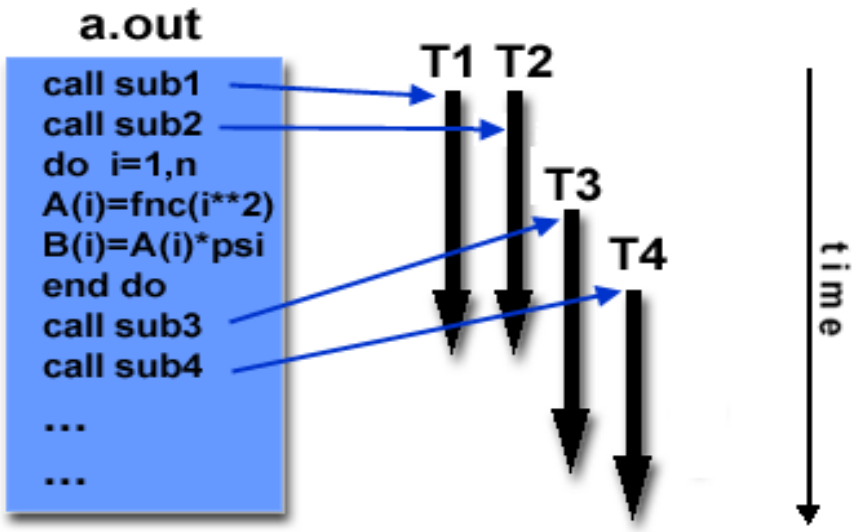
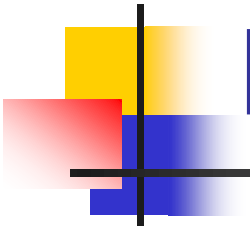


Network Programming

Network Programming

- Threads Model



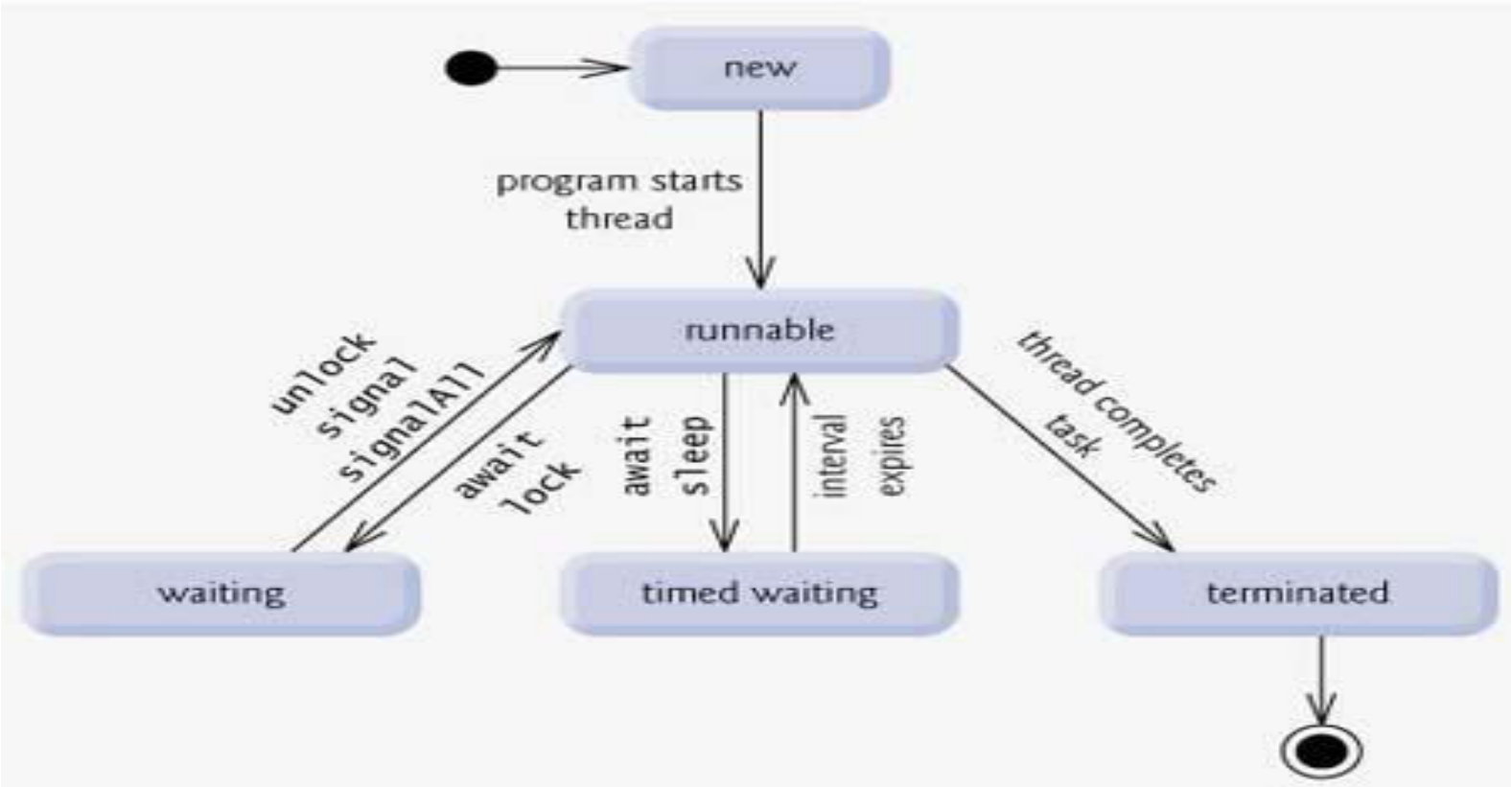


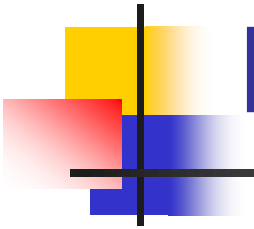
Network Programming

- Multithreaded programming, challenges include
  - Dividing work load
  - Overriding data
  - Data dependency
  - Deadlock
  - Testing and debugging

Network Programming

- Life Cycle of a Thread



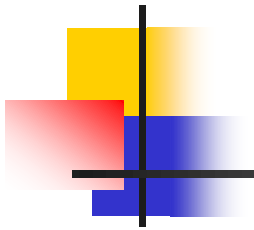


Network Programming

- Creating a Thread:
  - You can implement the Runnable interface.
  - You can extend the Thread class.

1 - Create Thread [Runnable]

```
class MyThread implements Runnable
{       Thread t;
        MyThread()
        {
          t = new Thread(this, “test Thread");
          System.out.println(“My thread: " + t);
          t.start();
        }
        public void run() {        …       }
}
```



```
class MyThread implements Runnable {
    Thread t;
    MyThread() {
                t = new Thread(this, “First Thread");
                System.out.println(“ My thread: " + t);
                t.start();
              }
    public void run() {
          try {
                for(int i = 5; i > 0; i--) {
                        System.out.println(“My Thread: " + i);
                        Thread.sleep(500);
                }
              } catch (InterruptedException e) {
                        System.out.println(" My interrupted.");
                }
          System.out.println("Exiting my thread.");
     }
}
```

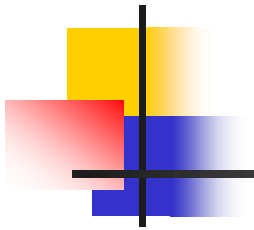
```
class MainThread
{
    public static void main(String args[])
    {
        new MyThread();                    // create a new thread
        try {
            for(int i = 5; i > 0; i--)
            {
                System.out.println("Main Thread: " + i);
                Thread.sleep(1000);
  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

- **output**

```
My thread: Thread[First Thread,5,main]
Main Thread: 5
My Thread: 5
My Thread: 4
Main Thread: 4
My Thread: 3
My Thread: 2
Main Thread: 3
My Thread: 1
Exiting my thread.
Main Thread: 2
Main Thread: 1
Main thread exiting.
```

2 - Extend Thread [extends]

```
class MyThread extends Thread
{
        MyThread()
       {
          super("test Thread");
          System.out.println("My thread: " + this);
          start();
        }
        public void run() {        …       }
}
```



```
class MyThread extends Thread {

        MyThread() {
                        super("Second Thread");
                        System.out.println(" My thread: " + this);
                        start();
                    }
        public void run() {
                try {
                        for(int i = 5; i > 0; i--) {
                                System.out.println("My Thread: " + i);
                                Thread.sleep(500);
                        }
                    } catch (InterruptedException e) {
                                System.out.println(" My interrupted.");
                      }
                System.out.println("Exiting my thread.");
         }
}
```

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

- **output**

```
My thread: Thread[Second Thread,5,main]
Main Thread: 5
My Thread: 5
My Thread: 4
Main Thread: 4
My Thread: 3
My Thread: 2
Main Thread: 3
My Thread: 1
Exiting my thread.
Main Thread: 2
Main Thread: 1
Main thread exiting.
```

Write a *Java program* to print *THIS IS A JAVA PROGRAM 10* times using functions?

```
class  MyJava  {
        public void print( ) {
                for(int i = 0 ; i < 10 ;  i++ )
                {
                        System.out.println(" this java is running ... " + i);
                }
        }
}
public class AClass {
        public static void main(String [] args  ) {
                MyJava j = new MyJava( );
                j.print( );
        }
}
```

Write a *Multithreaded Java program* to print *THIS IS A JAVA PROGRAM 10* times using functions?

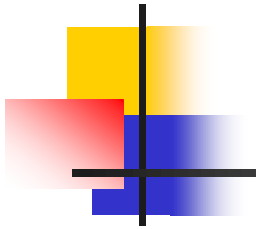
```
class  MyThread extends Thread {
        public void run( ) {
                for( int i = 0 ; i < 10 ;  i++ )
                {
                        System.out.println(" this thread is running ... " + i);
                }
        }
}
public class AClass {
        public static void main(String [] args  ) {
                MyThread t = new MyThread( );
                t.start( );
        }
}
```

Write a *Java program* to use a recursive function to calculate factorial?

```
public class BClass
{
    public static void main(String[] args) {
            int x = 5;
            int answer;
            answer =  factorial(x);
            System.out.println("The Answer " + x + " is " + answer);
       }
    static int factorial(int number)
    {
            if(number <= 1)
                        return 1;
            return number * factorial(number - 1);
    }
 }
```

Write a Multithreaded *Java program* to use a recursive function to calculate factorial?

```
class  MyThread implements Runnable
{
    Thread t;
    int num;
    MyThread(int num) {
            this.num=num;
            t = new Thread(this, "My Thread");
            t.start();
    }
    int factorial(int number)
    {
        if(number <= 1)
                return 1;
        return number * factorial(number - 1);
     }
    public void run( ) {
        int answer;
        answer =  factorial(num);
        System.out.println("The Answer"
         + num + " is " + answer);
    }
}
public class BClass
{
    public static void main(String[]
                                        args)
    {
            int x = 5;
            new MyThread(x);
     }
}
```

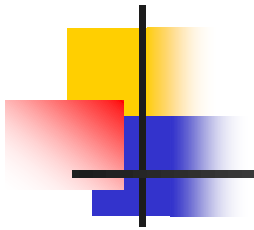


```
public class MyRunnable implements Runnable {

    public void run() {
        System.out.println("My thread!");
    }

    public static void main(String args[]) {
        (new Thread(new MyRunnable())).start();
    }

}
```



```
public class MyThread extends Thread {
    public void run() {
        System.out.println(" My thread!");
    }
    public static void main(String args[]) {
        (new MyThread()).start();
    }
}
```