

Multi Threading (~~Multi Tasking~~)

A multi thread program contain two or more parts that can run concurrently (simultaneously). Each part of such program is called thread and each thread define a separate path of exe.

Multi threading - Multi threading is a ability of an operating system to execute the different parts of the program called thread simultaneously. There is no interfere in any threads.

Advantage

- Communicate over a network.
- Performance take large amount of time.
- Distinguish tasks of varying priority.
- Allow the user interface to remain responsive.

Disadvantage

- App domain objects, and threads.
- Keeping track large no of threads consume.
- Controlling code execution.
- Destroying thread.

Working with threads :- The classes and interface in the System.Threading namespace provide a multithreading support in the .Net platform.

Creating and Running threads.

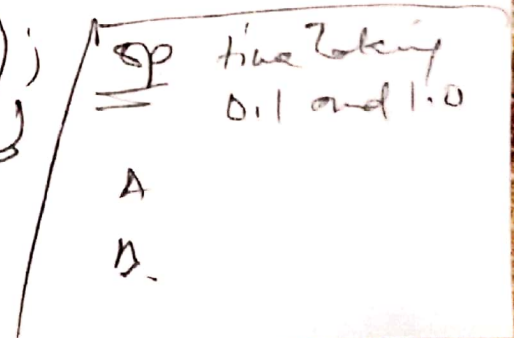
To create thread first of all create new instance

```
Thread myThread = new Thread (new ThreadStart  
(myfunc));
```

Prog. for threading construct

class program

```
{ static void main ()  
{  
Two thread creates { Thread thread1 = new Thread (new ThreadStart  
start (A));  
Thread thread2 = new Thread (new ThreadStart  
start (B));  
Thread1.start ();  
Thread2.start ();  
Thread1.join ();  
Thread2.join ();  
}  
static void A ()  
{ Thread.sleep (100);  
Console.WriteLine ("A");  
}  
static void B ()  
{ Thread.sleep (1000);  
Console.WriteLine ("B");  
}  
}
```



Thread States

(10)

- ↳ Unstarted state
- ↳ Running state
- ↳ Stopped
- ↳ Suspended.

Error handling (Exception handling)

When a program has a bug we can intercept it in the flow of execution of a piece of code. We have to first wrap it in a "try" block and then specify "catch" block matching that type of exception.

```
Ex  
try { int zero = 0;  
    res = (num/zero);  
}
```

```
catch (System.DivideByZeroException e)  
{  
    Console.WriteLine("Error: an attempt to  
    divide by zero");  
}
```

NOTE: - for multiple catch blocks, to catch different type of exception.

Exception handling program

```
class OutOfRange : Exception  
{  
}
```

```
class Demo
```

```
{ int n;  
  public int[] array;  
  public Demo (int n)
```

```
{ this.array = new int[n];  
  this.n = n;
```

```
}  
public void showElement (int i)
```

```
{ try { if (i <= 0) throw (new OutOfRange ());
```

```
catch (Exception e)
```

```
{ console.WriteLine ("Exception: {0}", e);
```

```
  console.WriteLine (array[i]);
```

```
}
```

```
} class test
```

```
{ public static void main ()
```

```
{ Demo test = new Demo (3);
```

```
  test.array [1] = 2;
```

```
  test.array [2] = 1
```

```
  test.showElement (0);
```

```
}  
}
```