



More on Methods

- **Overloaded methods**
- **Recursion**



Overloaded Methods

‘Overloading’ - A means of defining *different versions of a method* according to argument type(s) and/or return type

Example: method to add two values of the same type together



```
class Demo
```

```
{
```

```
    public static void Main( string[ ] args)
```

```
    {
```

```
        double result = Demo.add( 3.0, 4.0 );
```

```
        Console.WriteLine(“Result is “ + result );
```

```
    }
```

```
    public static double add( double a, double b )
```

```
    {
```

```
        double result = a + b;
```

```
        return result;
```

```
    }
```

```
}
```



Now we want the method to also be able to add two strings together

- Write a second method *also* called '**add**'

```
public static string add( string x, string y )  
{  
    string result = x + y;  
    return result;  
}
```



```
public static void Main( string[ ] args)
{

    Console.WriteLine("Result is " + Demo.add( 3.0, 4.0 ) );
    Console.WriteLine("Result is " + Demo.add( "first", "second") );
}
```

Result is 7.0

Result is firstsecond

JVM knows which method to call because
it compares *what is sent and returned*
i.e. it compares the method usage with the
method ***signatures***



Recursion

REPETITION CAN ALSO BE ACHIEVED BY **RECURSION**

```
while( val == true )  
{  
    do something;  
    change val;  
}
```

OR

```
method( val )  
{  
    if ( val == true )  
        do something;  
    else  
    {  
        change val  
        method( val )  
    }  
}
```



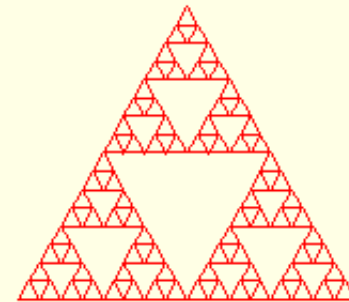
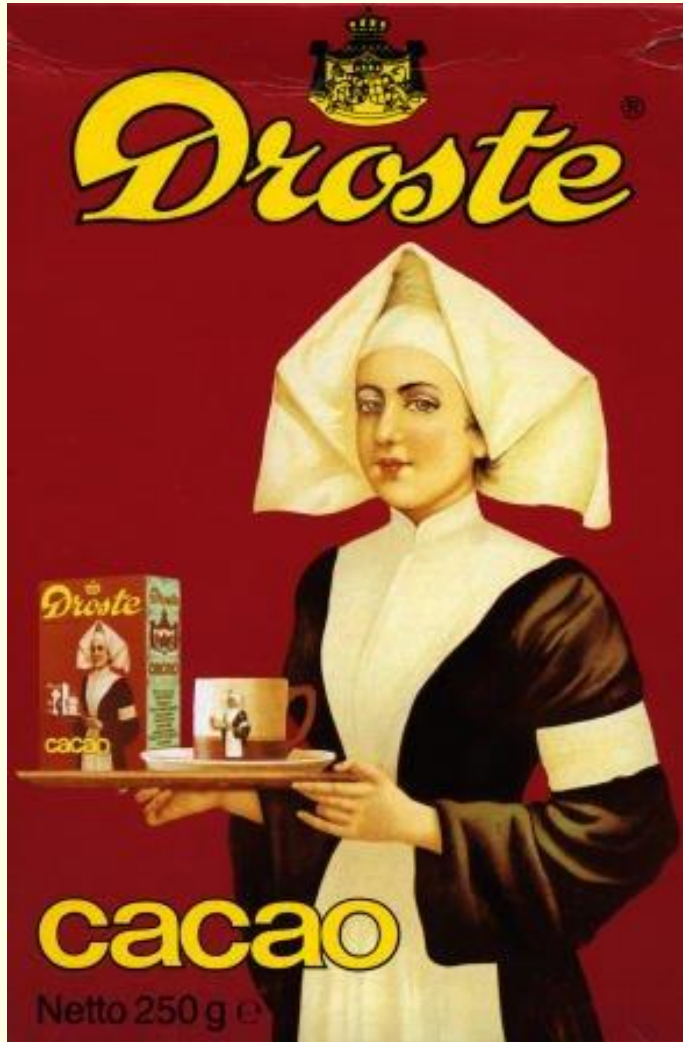
Anglia Ruskin
University

Rubick's Cube



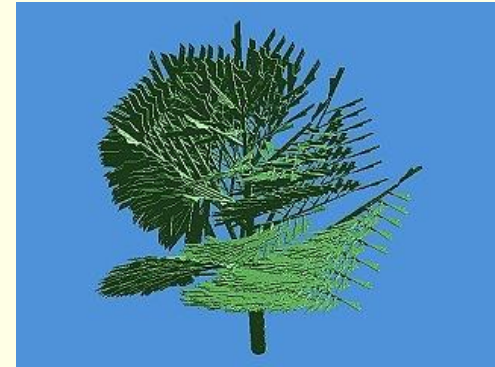
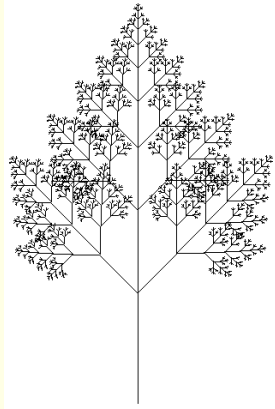


Anglia Ruskin
University



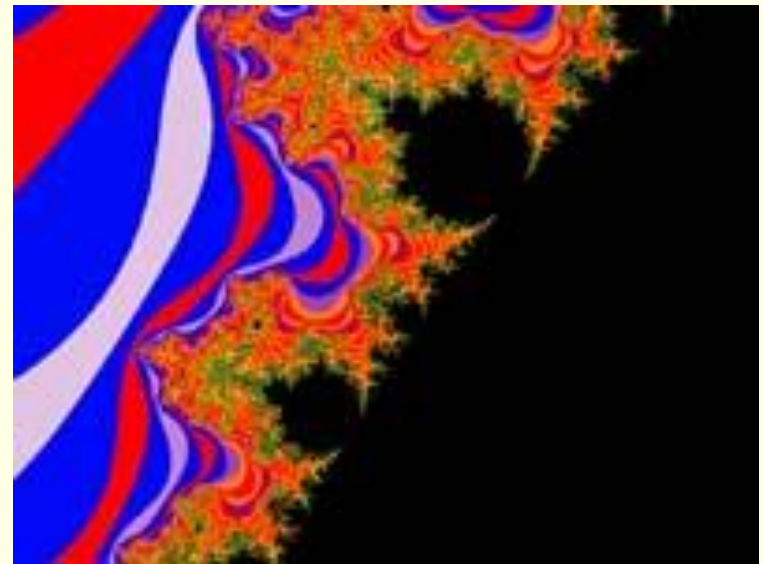
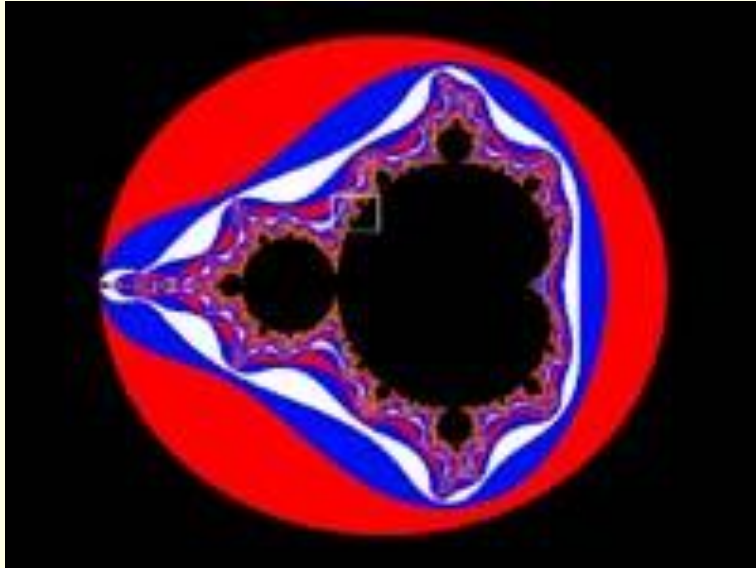


Anglia Ruskin
University



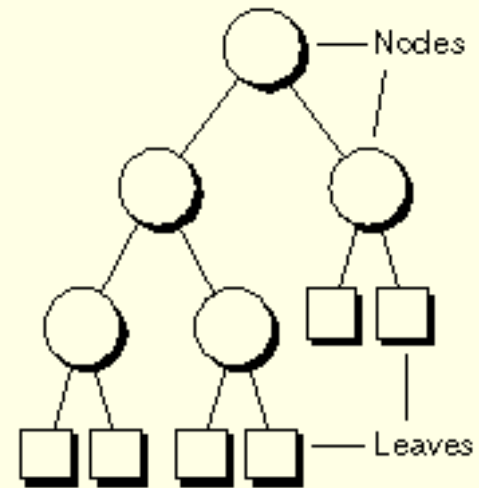


Anglia Ruskin
University





Anglia Ruskin
University



Binary Tree



Conclusion

- Overloaded methods allow variations of a method to be defined, each with the same name but with different arguments
- A recursive method is a method that calls itself, and is an alternative way of implementing repetition