

Classes

A class is a "module" of a Java program

Typically 1-1 mapping between Class and file

```
class Counter
{
    // Code for class goes here !
}
```

A Class describes a particular type of Object

Objects of that type are then created at run time

Objects are "instances" of Class that describes them

Convention is that Class names start with a capital

Java Types

Sorry, but this next bit is a little confusing...

The majority of data types in Java are Classes

The exception to this are "Primitive" types
(int, char, boolean, float etc)

Primitives are just simple data (the same as in C)
Everything else in Java is a Class !

Java provides the concept of an array (just like C)
These can contain either Primitives or Class types

Attributes

A Class has a number of data fields or "Attributes"

These are global to (accessible within) the class

But importantly NOT global to the whole program

(Remember the notion of "Encapsulation" ?)

For example:

```
class Counter
{
    int count;
}
```

Methods

Each Class has a bunch of functions: "Methods"

```
class Counter
{
    int count;

    void increment() {
        count++;
    }
}
```

Such Methods are "attached" to that Class

They are called "on" a specific instance of that Class

```
Counter carCounter;
carCounter.increment();
```

Difference Between Functions & Methods

Methods are tied to a particular object

Functions are just "floating around"

In C you just call a function in isolation:

```
printf("Hello");
```

In Java you call a method ON a particular Object:

```
out.print("Hello");
```

```
serial.print("Hello");
```

```
file.print("Hello");
```

Constructor Methods

Special methods exist to create instances of a Class
(Remember that such instances are called Objects)
Constructors have the same name as the Class:

```
class Counter
{
    int count;

    public Counter() {
        count = 0;
    }
}

Counter myCounter = new Counter();
```

Notice there is no return type with a constructor !

Multiple Constructors

We can have simple constructors with default values:

```
public Counter() {  
    count = 0;  
}
```

Or complex ones, where we passing in the settings:

```
public Counter(int startValue) {  
    count = startValue;  
}
```

Providing more than one method with the same name like this is referred to as "Overloading"