

9.5 The 'main' Method

Hopefully after reading this chapter, the line you've been writing for weeks now should make a bit more sense:

```
public static void main(String[] args)
```

We still need to deal with `public` and `static`, which we'll do later in the module, but:

```
void main(String[] args)
```

should now be clear – this is a method, called `main` with one argument which is an array of Strings. The convention in Java (and other languages like C, C++ and C# - most widely-used modern languages) is that when you run a program, the system looks for a method called `main` and starts the program there. The `main` method is just an ordinary method like any other, and it's possible to pass in parameters as an array of Strings. Here's a program that deals with parameters:

```
public class SquareArgs {
    public static void main(String[] args) {
        if (args.length == 0) {
            System.out.println("Supply a number");
        } else {
            int value = Integer.parseInt(args[0]);
            System.out.println("Square " +
                               square(value));
        }
    }

    static int square(int num) {
        return num * num;
    }
}
```

The if statement first checks to see if there are any arguments because if `args.length==0` then the array `args` is empty and the user hasn't typed any values in. Otherwise we:

- Assume that the first thing they've typed is the argument and ignore anything else they might have written after it (`args[0]` will contain whatever they typed first as an argument – remember that arrays in Java start with element zero).
- Use `parseInt` to turn the string into an integer – this will crash if the user types a non-integer, which is not great but this is meant to be a simple example.

The way we 'call' the `main` method with parameters is slightly different to the way we would call it within a program and we don't use brackets:

Type this on the command line - the output is the line in bold:

java SquareArgs 5

Square 25

if we leave off the arguments:

java SquareArgs

Supply a number

if we type extra numbers, they are ignored:

java SquareArgs 7 3

Square 49

Here's a video explaining and showing this:

