THE JAVA CLASS CONSTRUCTOR

When we make field variables private, we would need another way to assign values to them. One way we could do this is with something called a *constructor*. This is a method you can use to set initial values for field variables. When the object is created, Java calls the constructor first. Any code you have in your constructor will then get executed. You don't need to make any special calls to a constructor method, they happen automatically when you create a new object.

Constructor methods take the same name as the class; like this:

```
package exams;
public class StudentResults {
   private String Exam_Name;
   private String Exam_Score;
   StudentResults () {
      //class constructor
   }
}
```

The name of the constructor is StudentResults. This is exactly the same name as the class itself. Unlike normal methods, class constructors do not need a return type like int or double, nor any return value. You can, however, pass values over to your constructors. If we want to pass values over to our field variables, we can do this:

```
StudentResults (String name, String score) {
   Exam_Name= name;
   Exam_Score= score;
}
```

Here we've added two String variables to the round brackets of the constructor. Inside the curly brackets we've assigned these values to the Exam_Name and Exam_Score fields. When you create a new object, you'd then need two strings between the round brackets of the class name:

```
StudentResults aStudent= new StudentResults
("Maths", 83)
```

When the object is created, the values “Maths” and “83” will be handed over to the constructor.

However, it is a good idea to just set some default values for your field variables. These values will then be assigned when the object is created.