Figure 8.0 Constructors revisited.

In the sequence

```
Student stu1 = new Student();    // [1]
stu1.setName("Richard");         // [2]
```

there is a "brief period of time" between statements [1] and [2] where **stu1** has no name. It would make more sense to give **stu1** a name as it is created.

```
Student stu1 = new Student("Richard");
```

To do this, we need to add a *constructor* to the **Student** class:

```
public Student (String newName) {
    // no return type … and its name is the same as the class name
    name = newName;
}
```

Constructor overloading: We can add more than one constructor, providing the ______________________ differ:

```
public Student (String newName, String newNumber) {
    name = newName;
    stunum = newNumber;
}
```

This second constructor is invoked when we type:

```
Student stu1 = new Student("Richard", "890792492");
```

Exercise: After adding explicit constructors to a class, does the default constructor still exist? For example, after adding the above two constructors to the **Student** class, can you still say

```
Student stu1 = new Student();
```

Try it!