Q34 - Programmatic Servlets Addition

Consider the following Servlet code and the `ServletContainerInitializer` code.

com.nullhaus.MyJar1Servlet

```
package com.nullhaus;

import javax.servlet.annotation.*;
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;

@WebServlet(value = "/foo/*", name="NullHaus1")
public class MyJar1Servlet extends HttpServlet {
}
```

com.nullhaus.MyInit

```
package com.nullhaus;

import javax.servlet.*;
import java.util.*;

public class MyInit implements ServletContainerInitializer {
    public void onStartup(Set<Class<?>> c, ServletContext ctx) throws ServletException {
        try {
            Class klass = Class.forName("com.nullhaus.MyJar1Servlet");
            Class<MyJar1Servlet> clazz = (Class<MyJar1Servlet>)klass;

            Servlet s = ctx.createServlet(clazz);
            ServletRegistration.Dynamic d = ctx.addServlet("NullHaus2", s);

            d.addMapping("/baz/*");
        } catch (ClassNotFoundException e) {
            // ...
        }
    }
}
```

Assume that the `MyInit` class is properly registered in the container as a `ServletContainerInitializer`. Choose the statements that are true:

a. There will be at least one instance of the `MyJar1Servlet` named NullHaus1

b. There will be at least one instance of the `MyJar1Servlet` named NullHaus2

c. There will be exactly two instances of the `MyJar1Servlet` named NullHaus1 and NullHaus2 respectively

d. A runtime exception will be thrown

e. This code doesn't compile

Hide answer

a,b

Reference: page 62, 8.1.1 "@WebServlet"

Explanation: When the same servlet class -- but with different name -- is instantiated using programmatic addition, **the container will create two instances of the servlet**. The annotated one will have configuration as defined using the annotations, and the programmatic one will have its own configuration.