**Introduction:**

Use format specifiers in multiple ways to determine the results. The first format specifier is /%f with an int. The second /%d with a double, and lastly, show one value with two specifiers.

| | | | | | |
|---|---|---|---|---|---|
| FID001 | **I want to** | display an int using /%f | **so that** | I can see the result | student |
| FID002 | **I want to** | display a double using /%d | **so that** | I can see the result | student |
| FID003 | **I want to** | display a value using to format specifiers | **so that** | I can see the result | student |

**Body**

When /%f is used with an integer, f != java.lang.Integer; appears. On the other hand, d != java.lang.Double when /%d is used with a double. When two specifiers are used for one value different outcomes appear for what specifiers were used. For example, the first specifier gets applied first. If it's true, then it continues to the next. However, the second one causes an error to the operation.

**Conclusion**

If a specifier is used that's redundant to the variable there is an error; !=java.lang.data. If two specifiers are used on one variable then the first one is used and the second is an error. Even if both are true.

I know only the specifiers that were in the chapter, so my testing of multiple specifiers was very limited.