Software Engineering II ANSWERS IN BLUE!!

Aspect Oriented Programming **Tutorial 5**

Question 1

Explain the following code in detail by commenting each line

```
public aspect EL {
        pointcut scope(): within(uk.ac.liv.csc.comp319.common.*);
// Defines joint points to be any code running in package classes
uk.ac.liv.csc.comp319.common.

        pointcut pr():
                get(* System.out) ||
                get(* System.err) ||
                call(* Throwable+.printStackTrace());
// Defines joint points to be any read attribute of System.out or
System.err or calls to printStackTrace();


        declare warning : scope() && pr(): "";     // Forces a warning of use
of console calls in code in given package
}
```

Question 2 What is the different between a pointcut and a joinpoint?

Joint point is the place in the target code the advice code is injected

pointcut **Defines a set of joinpoints using a pattern definition**

Question 3 Download the aspect source code examples from tinyurl.com/comp319.
Look at the DobValidate.aj
Explain the purpose of the around and proceed keywords?

The around key word, means that the original target code is skipped

When using the around keyword, you can revert to calling the original code, by using the proceed keyword.

This means the Aspect can control the execution of the target code, deciding if to skip or execute it

Question 4 What are the benefits/drawbacks in applying AOP for the examples in Q1 and Q3

Benefits **Benefits**

Easy and quick to apply control to whole application.

Drawbacks Arguably harder to test application.

Q3 Validation is arguably not a cross cut concern but should be coherent with the class definition.

Question 5 Define an Aspect for the following:
Log all setters (not method setters) for any class which is in the package comp.liv319.
What options are available for the pointcut position?

```
    pointcut pr():
        set(uk.ac.liv.csc.comp319.*(8))
```

before	before setter is called
after	after setter is called
around	before and allows skipping of setter

Question 6 Read and review the paper by Greenwood and Blair on autonomic systems.

a) Why did they think AOP applied to autonomics?
Autonomics is a cross cut concern involving the whole of the application, AOP is a cross cut approach

b) What are the alternatives they outlined?
Java hotswap (have a look in paper for alternatives)