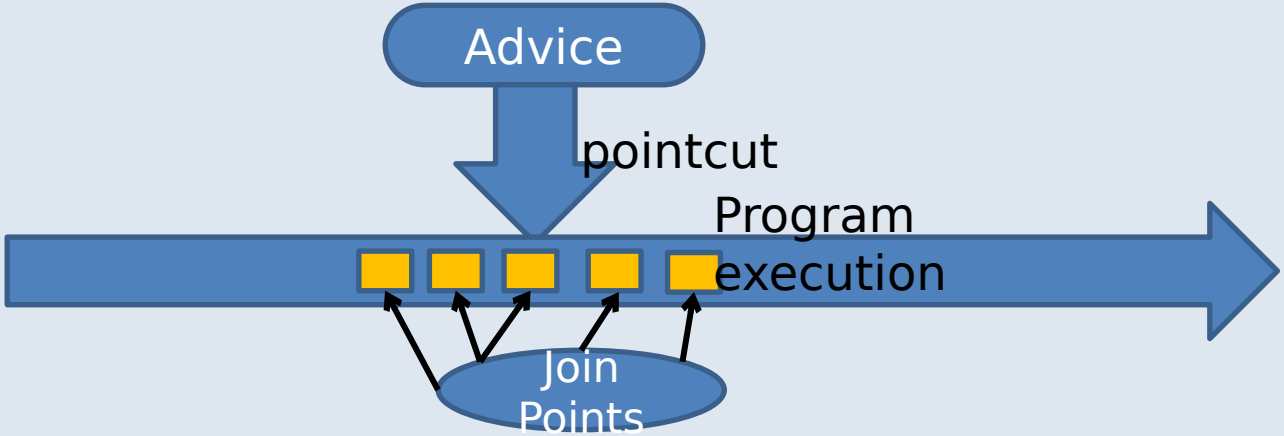
Aspect Oriented Programming

- AOP is a programming paradigm, like OOP
- OOP fails in the modularization of cross-cutting aspects
  - Security
  - Tracing
  - Benchmarking
  - Validation
  - Others
- AOP completes OOP in the cross-cutting aspects

AOP Concepts

- Join Point
  - *Point during the execution of a program, such as the execution of a method or the handling of an exception*
- *Advice*
  - Action taken by an aspect at a particular join point
- *Pointcut*
  - Predicate that matches join points. Advice is associated with a pointcut expression and runs at any join point matched by the pointcut



Advice Types

- Before advice
  - Advice that executes before a join point
- After returning advice
  - Advice to be executed after a join point completes normally
- After throwing advice
  - Advice to be executed if a method exits by throwing an exception
- **Around advice**
  - **Advice that surrounds a join point such as a method invocation. This is the most powerful kind of advice. Around advice can perform custom behavior before and after the method invocation. It is also responsible for choosing whether to proceed to the join point or to shortcut the advised method execution by returning its own return value or throwing an exception.**

Pointcuts Expression Examples

- Execution of any public method:
  execution(public * *(..))
- Execution of any method with a name beginning with "set":
  execution(* set*(..))
- Execution of any method defined by the AccountService interface:
  execution(* com.xyz.service.AccountService.*(..))
- Execution of any method defined in the service package:
  execution(* com.xyz.service.*.*(..))
- Execution of any method defined in the service package or a sub-package:
  execution(* com.xyz.service..*.*(..))

ProceedingJoinPoint Class

proceed()

Proceed with the next advice or target method invocation

Signature getSignature()

Returns the signature at the join

Object[] getArgs()

Returns the arguments at this join point.

Object getTarget()

Returns the reference to a target object