

---

# AOP Case studies

---

ASET (Fall 2013)

Amariei G. Daniel  
Software Engineering

---

# Contents

---

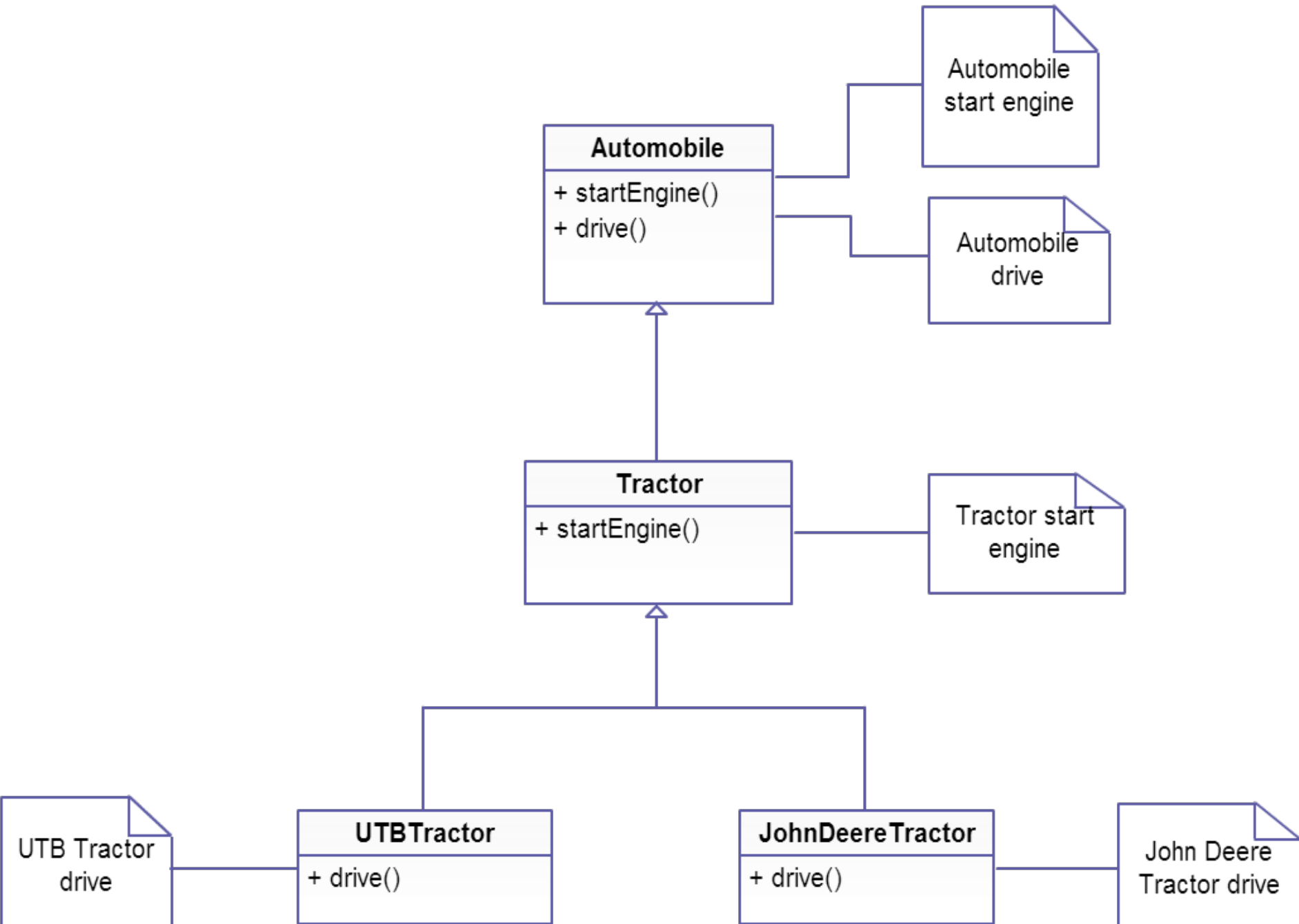
1. Using AOP with class hierarchies
  2. Runtime measurement:
    - a. AOP
    - b. The classical approach
-

# 1. Using AOP with class hierarchies

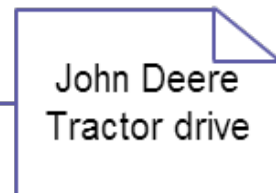
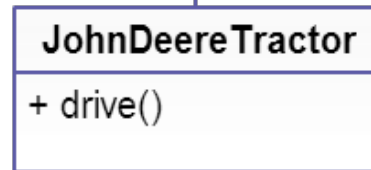
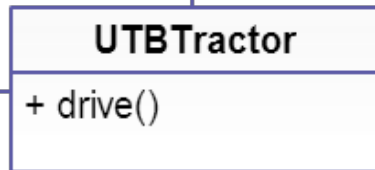
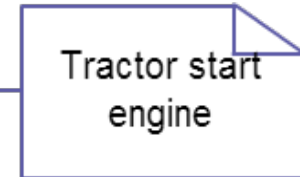
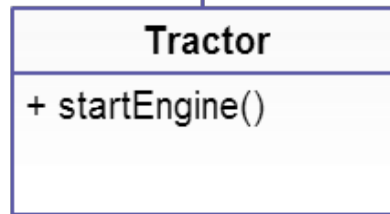
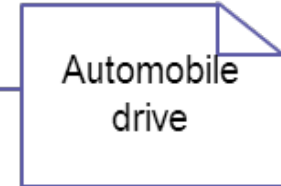
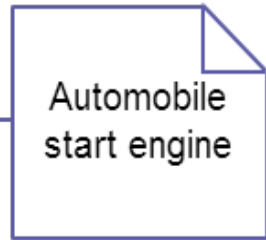
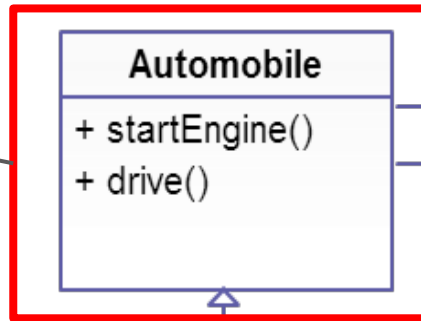
---

Add an aspect for the base class.  
Will it work with the derived classes?

---



Add aspect



# AutomobileBeforeAfterAspect

---

```
pointcut startEngine() :  
    target(Automobile) &&  
    call(public void startEngine());
```

---

# AutomobileBeforeAfterAspect

---

before() : startEngine() ...

after() : startEngine() ...

---

# AutomobileBeforeAfterAspect

---

```
pointcut drive() :  
    target(Automobile) &&  
    call(public void drive());
```

---



# AutomobileBeforeAfterAspect

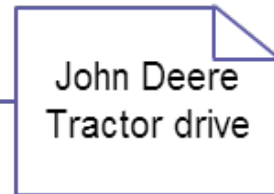
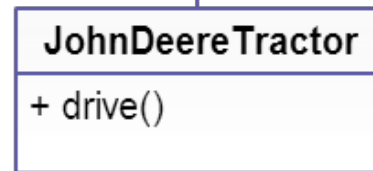
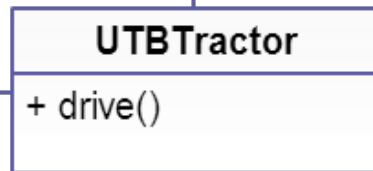
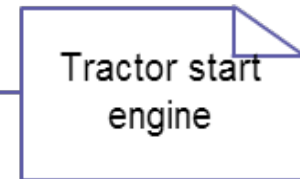
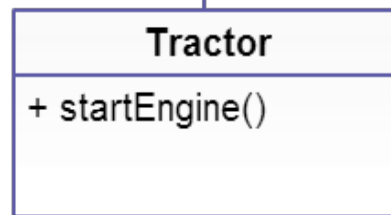
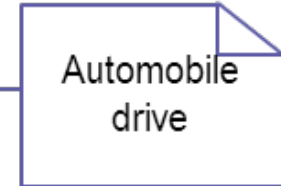
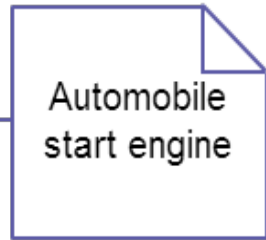
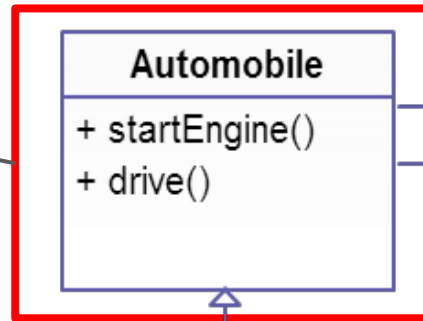
---

before() : drive() ...

after() : drive() ...

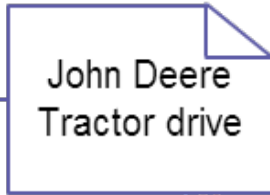
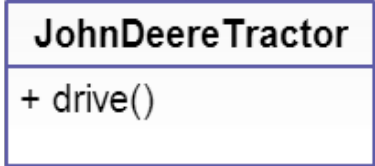
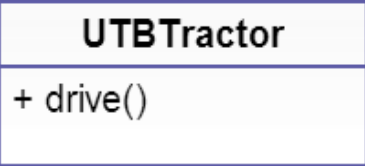
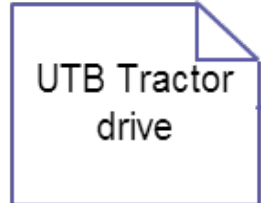
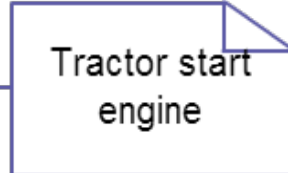
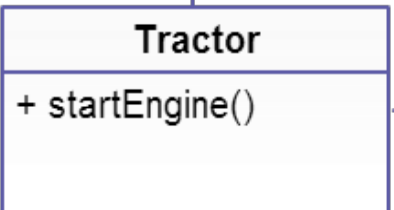
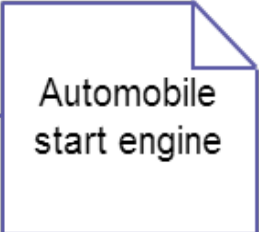
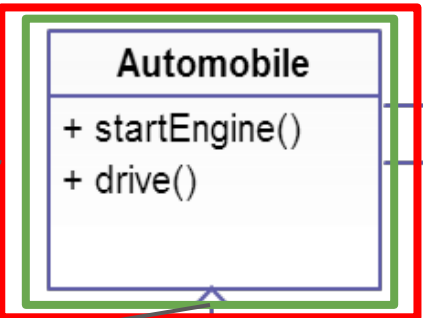
---

Aspect



Aspect

Object



# AOP in action (Automobile)

---

```
Automobile automobile = new Automobile();  
automobile.startEngine();  
automobile.drive();
```

---

# AOP in action (Automobile)

---

[Aspect]Before starting the engine of an automobile  
call(void automobile.Automobile.startEngine())

**Automobile start engine.**

[Aspect]After starting the engine of an automobile  
call(void automobile.Automobile.startEngine())

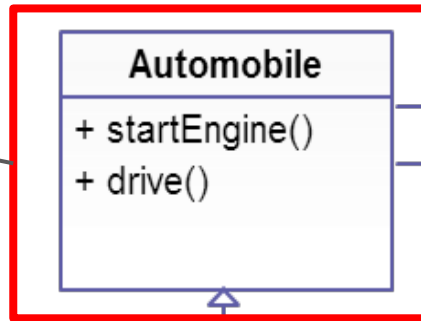
[Aspect]Before driving an automobile  
call(void automobile.Automobile.drive())

**Automobile drive.**

[Aspect]After driving an automobile  
call(void automobile.Automobile.drive())

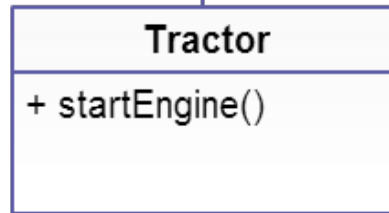
---

Aspect

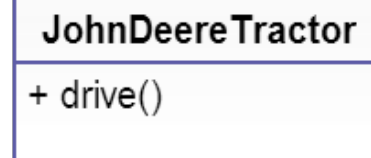
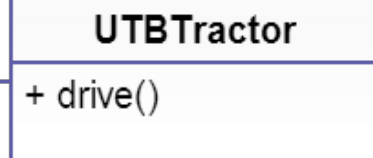


Automobile start engine

Automobile drive



Tractor start engine



UTB Tractor drive

John Deere Tractor drive

Aspect

**Automobile**

+ startEngine()  
+ drive()

Automobile  
start engine

Automobile  
drive

Object

**Tractor**

+ startEngine()

Tractor  
start  
engine

**UTBTractor**

+ drive()

UTB Tractor  
drive

**JohnDeereTractor**

+ drive()

John Deere  
Tractor  
drive

# AOP in action (UTBTractor)

---

```
UTBTractor utbTractor = new UTBTractor();  
utbTractor.startEngine();  
utbTractor.drive();
```

---



# AOP in action (UTBTractor)

---

[Aspect]Before starting the engine of an automobile call  
(void automobile.UTBTractor.startEngine())

## **Tractor start engine**

[Aspect]After starting the engine of an automobile call(void  
automobile.UTBTractor.startEngine())

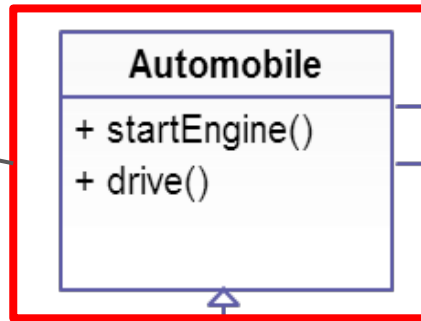
[Aspect]Before driving an automobile call(void automobile.  
UTBTractor.drive())

## **UTB Tractor drive**

[Aspect]After driving an automobile call(void automobile.  
UTBTractor.drive())

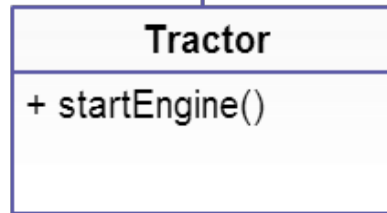
---

Aspect

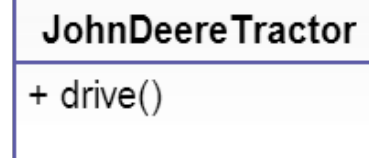
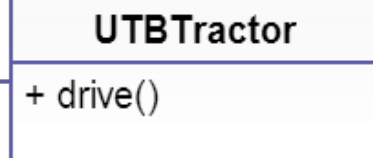


Automobile start engine

Automobile drive



Tractor start engine

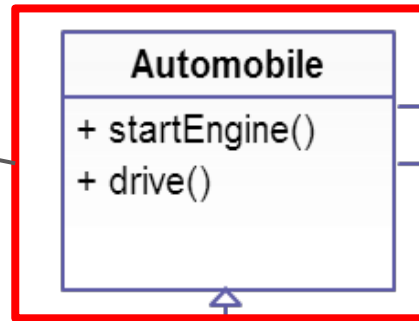


UTB Tractor drive

John Deere Tractor drive

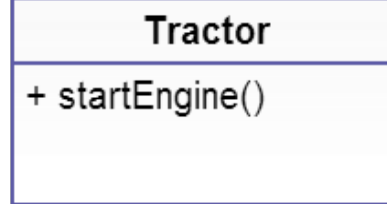
Aspect

Object



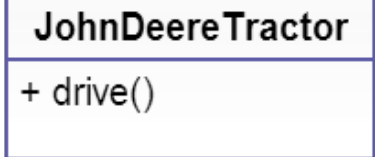
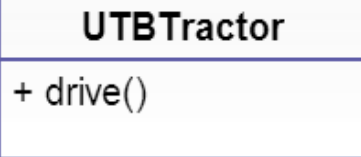
Automobile start engine

Automobile drive



Tractor start engine

UTB Tractor drive



John Deere Tractor drive

# AOP in action (JohnDeereTractor)

---

```
UTBTractor utbTractor = new UTBTractor();  
utbTractor.startEngine();  
utbTractor.drive();
```

---

# AOP in action (JohnDeereTractor)

---

[Aspect]Before starting the engine of an automobile call  
(void automobile.JohnDeereTractor.startEngine())

## **Tractor start engine**

[Aspect]After starting the engine of an automobile call(void  
automobile.JohnDeereTractor.startEngine())

[Aspect]Before driving an automobile call(void automobile.  
JohnDeereTractor.drive())

## **John Deere Tractor drive.**

[Aspect]After driving an automobile call(void automobile.  
JohnDeereTractor.drive())

---

## 2. Runtime measurement

---

**Ten rounds** of computing the Fibonacci numbers from **1 to 40**

Recursive approach

---

# AOP

---

Average time (nanoseconds):

**76133002.9375**

---

# The classical approach

---

Average time (nanoseconds):

**71829552.6925**

---