

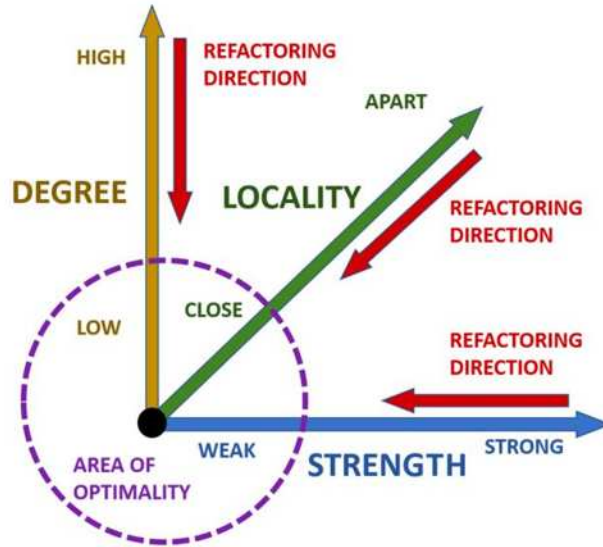
# Making sense of Connascence



*“the birth and growth of two or more things at the same time”*

# Connascence

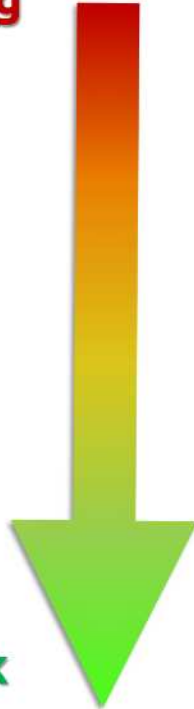
... is a software quality metric & taxonomy for different types of coupling



... give us an insight into the long-term impact of our code regarding flexibility

# Type of Connascence

**Strong**



**Weak**

- 10 **Manual task**
- 9 **Identity**
- 8 **Value**
- 7 **Timing**
- 6 **Execution order**
- 5 **Position**
- 4 **Algorithm**
- 3 **Meaning (Convention)**
- 2 **Type**
- 1 **Name**

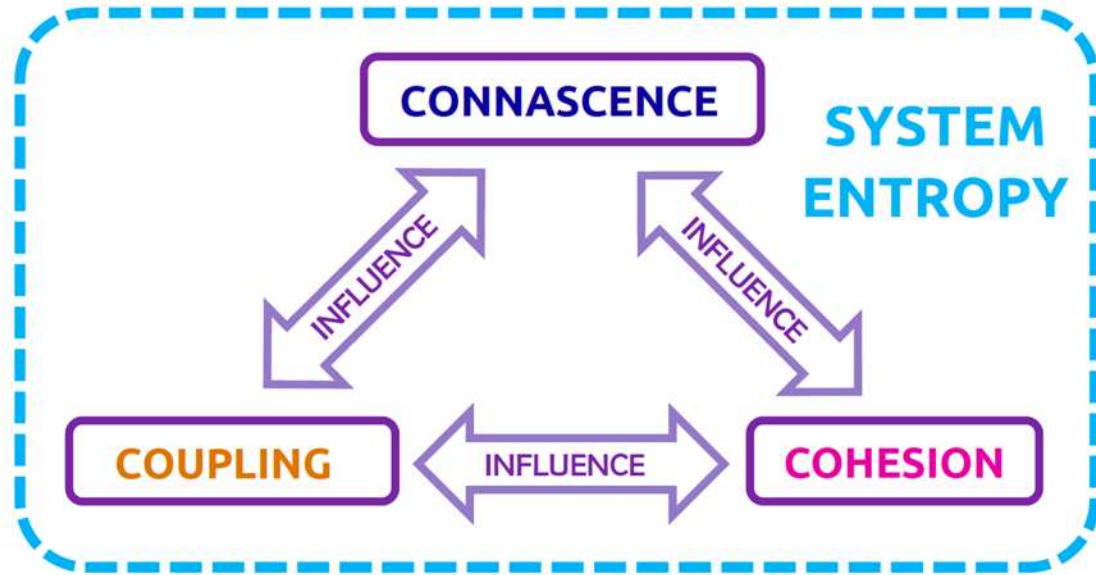
**Dynamic**  
(discoverable  
only at runtime)

**Static**  
(discoverable  
at compile time)



Lesson 2

TRAINING PROGRAMME



**ENTROPY** as the degree of disorder in a system

# Sample 1: Connascence of Timing

```
07:09
[...]
```



# Sample 2: Connascence of Execution Order

```
int main() {
    int x = 1;
    int y = 2;
    int z = 3;

    // ...

    // ...

    // ...

    // ...
}
```

**FAIL**

# Sample 3: Connascence of Position

```
public void setAge(int age) {
    this.age = age;
}

public void setAge(int age, String name, int sex) {
    this.age = age;
    this.name = name;
    this.sex = sex;
}

public void setAge(int age, String name, int sex, int height) {
    this.age = age;
    this.name = name;
    this.sex = sex;
    this.height = height;
}

public void setAge(int age, String name, int sex, int height, int weight) {
    this.age = age;
    this.name = name;
    this.sex = sex;
    this.height = height;
    this.weight = weight;
}
}
```





# Sample 4: Connascence of Meaning





# Conclusions

- It make sense to follow this metric to write better code
- Reduce coupling as much as possible
  - easier to change and extend the code
- It's not very easy to find them, but try hard!

# References

- Alcor Academy
- [connascence.io](https://connascence.io)

