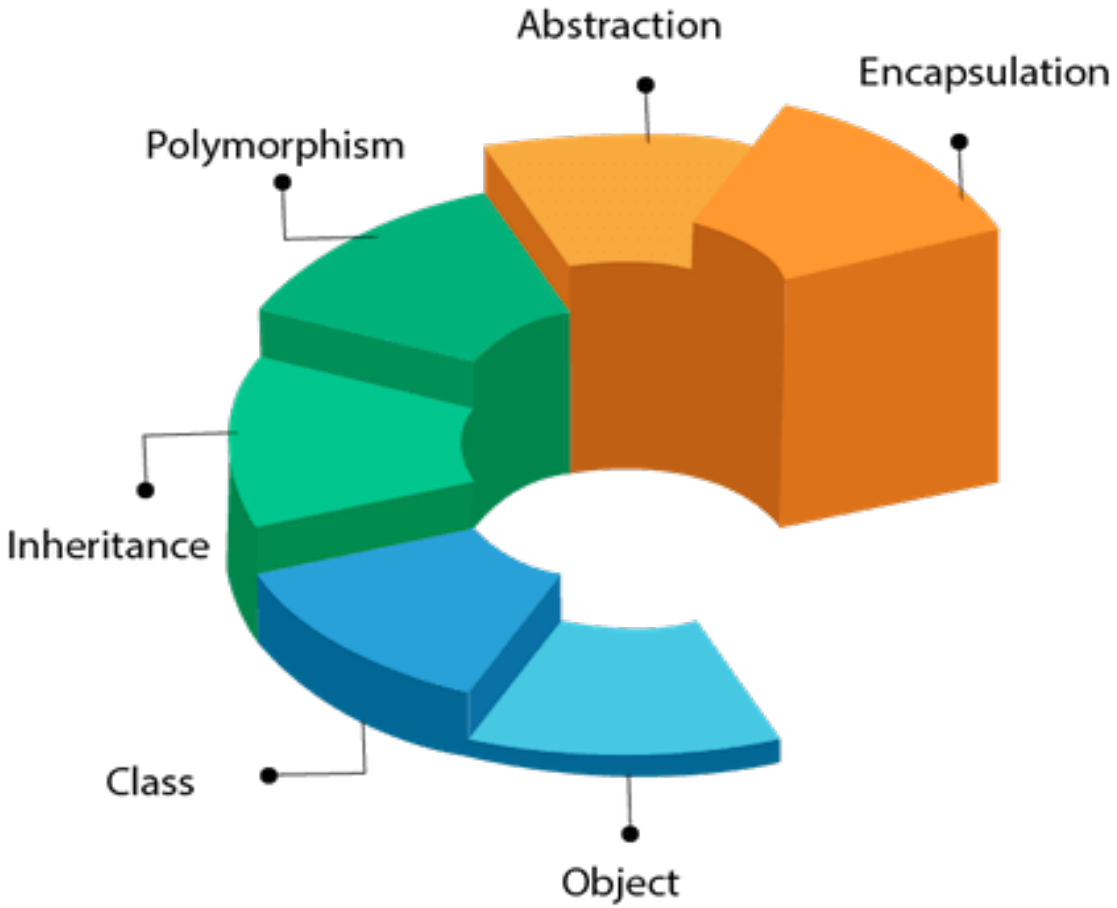
OBJECT-ORIENTED PROGRAMMING

AARHUS
UNIVERSITY
DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING

HENRIK BITSCH KIRK
ASSOCIATE PROFESSOR

OOPs (Object-Oriented Programming System)



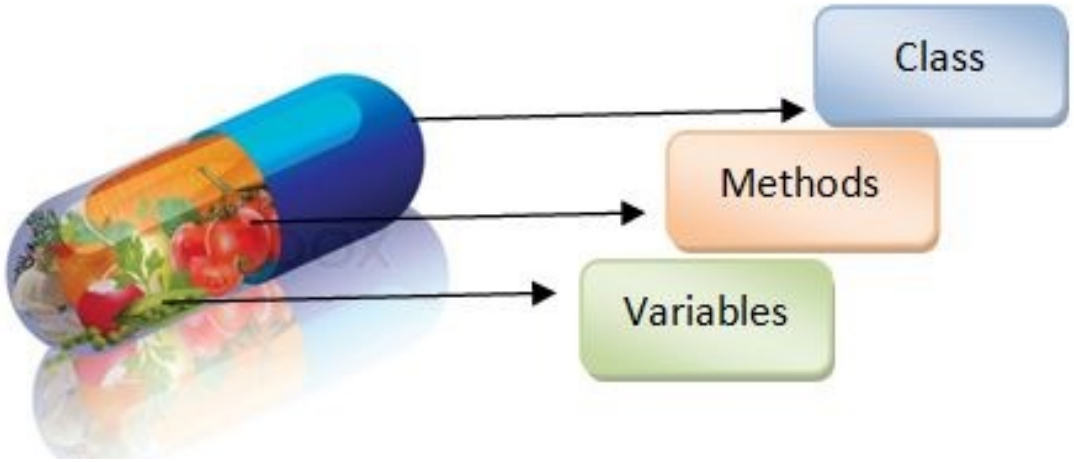
ENCAPSULATION

> Write shy code - modules that don't reveal anything unnecessary to other modules and that don't rely on other modules' implementations.
>
> - DAVE THOMAS

ENCAPSULATION

Discuss

- What are the benefits of encapsulation?
- What are the drawbacks?



"Details required to return a result from a class method should be hidden from the calling method"

EVIL CLIENT

EVIL CLIENT

```
public class TrackManager {
  public List<Track> tracks;
}
```

```
public class EvilClient {
  public void DestroyTracks(TrackManager tm) {
    tm.tracks.Clear();
} }
```

```
public class TrackManager {
  private List<Track> tracks;
  public Track GetTrack(string tag);
}
```

```
public class EvilClient {
  public void DestroyTracks2(TrackManager tm) {
    tm.GetTrack("A323").tag = "PWND";
    tm.GetTrack("PWND").course = -1000;
} }
```

```
public class TrackManager {
  private List<Track> tracks;
  public void SetSpeed(string tag, int s
  public void SetCourse(string tag, int
course);
}
```

```
public void DestroyTracks3(TrackManager tm) {
  tm.SetCourse("A323", -1000);
}
```

ABSTRACTION

> The purpose of abstraction is not to be vague, but to create a new semantic level in which one can be absolutely precise
>
> - EDSGER DIJKSTRA

ABSTRACTION

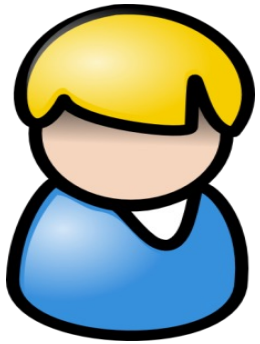
Discuss

- What does abstraction mean?

ABSTRACTION – AN EXAMPLE

Lecture:
- Name
- Hand-ins
- Group number

Office:
- AU-id
- Study program
- Passed courses



Bob the student

Student:
- Name
- Group number
- Active courses

Friend:
- ...

ORBIT Lab:
- ...

- These are all different abstractions of Bob.
- Which one to use depends entirely on the context in which Bob is represented.
- Creating the right abstraction of entities is very important to your design. Include the relevant, exclude the irrelevant.

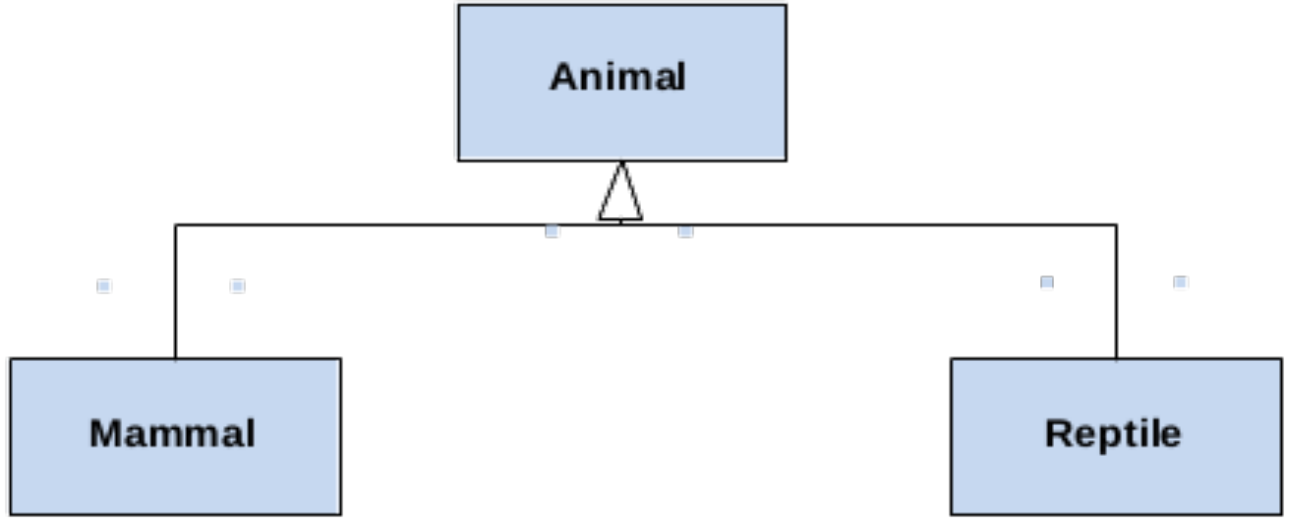
INHERITANCE

> There is nothing wrong with inheritance, or even multiple inheritance. It's a language feature just like any other. It can be used wisely or foolishly
>
> - ROBERT C. MARTIN

INHERITANCE

Discuss:

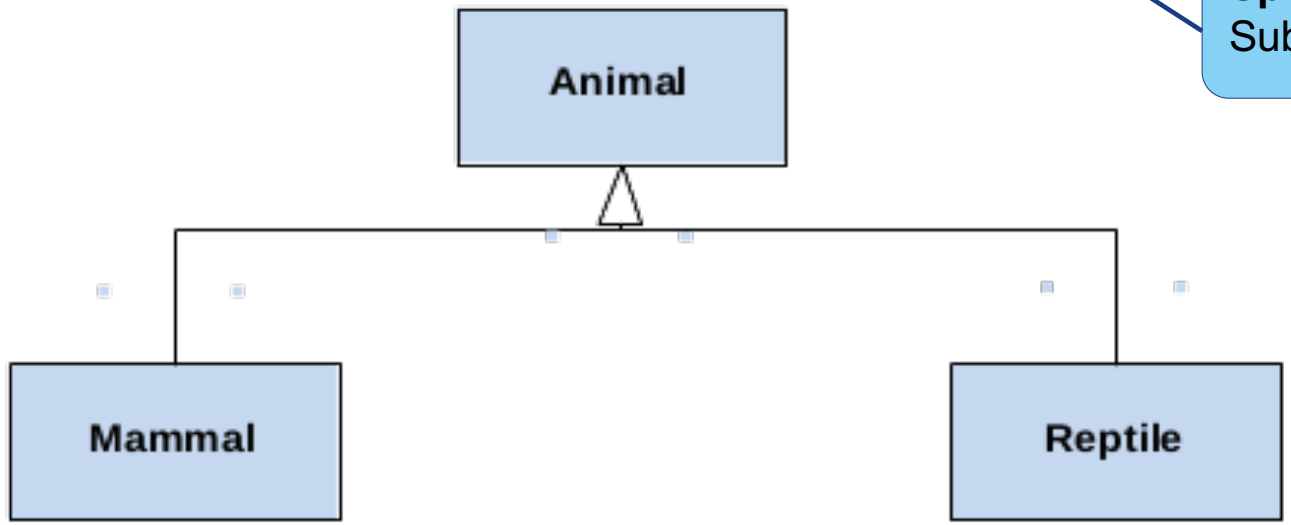
- Inheritance is commonly described as an “is-a relationship” between classes – how do you understand this?
- Explain the concepts “specialization” and “generalization” wrt. inheritance.



INHERITANCE

- Mammal inherits from Animal means…
  - Everywhere an object of class Animal is used, an object of class Mammal can be used (“compile-time” or “is-a” inheritance)
  - Every property that is desirable for clients of class Animal must be the same for class Mammal (“run-time” or “behaves-like” inheritance)

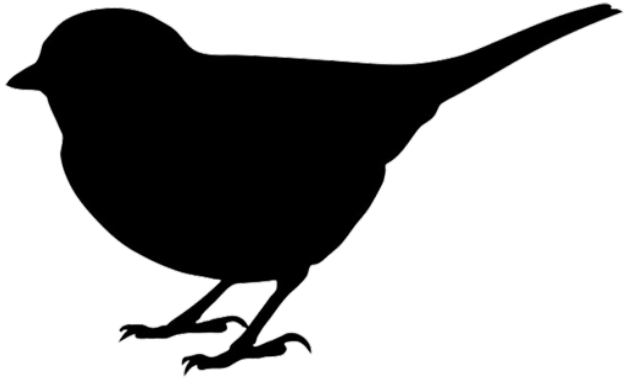
Spoiler: This is Liskov’s Substitution Principle



INHERITANCE – A QUIZ

Your turn: In 10 minutes...

1. Arrange the classes Bird, Penguin and Swallow in an inheritance hierarchy
2. Suppose you must add a method LayEgg() to Bird. Does your hierarchy still hold?
3. Suppose you must add a method Fly() to Bird. Does your hierarchy still hold
4. If the hierarchy breaks: How will you handle it?



POLYMORPHISM

> So much complexity in software comes from trying to make one thing do two things.
>
> - RYAN SINGER

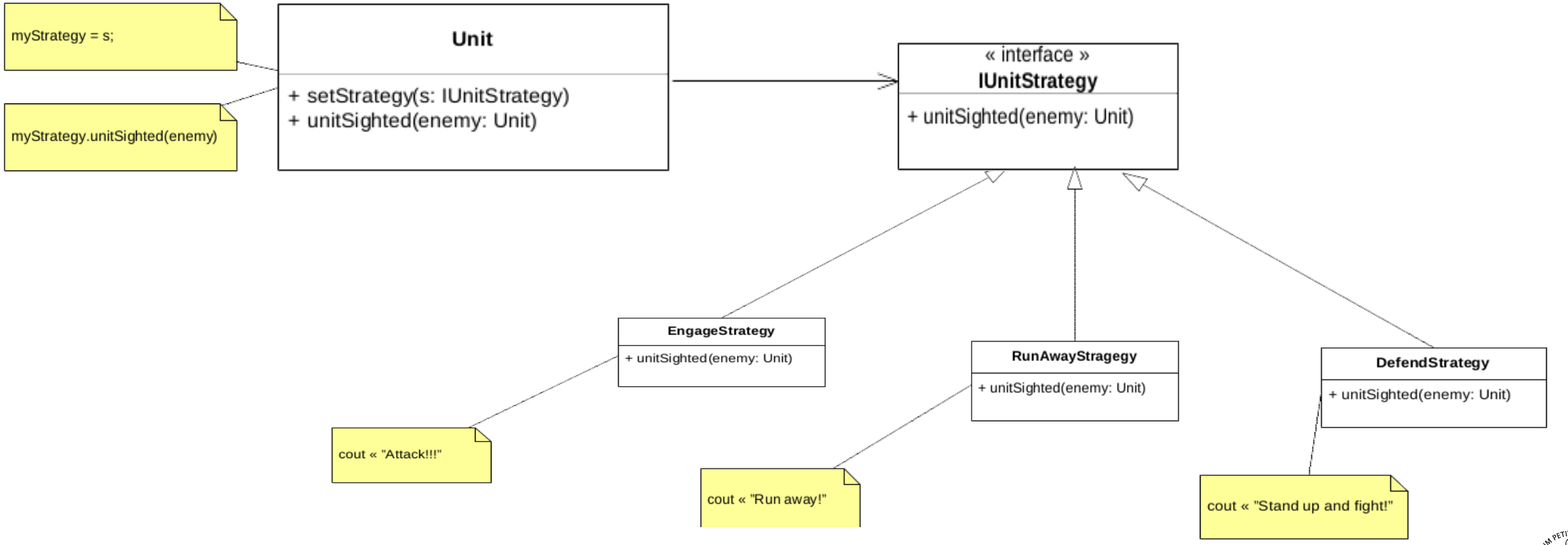
POLYMORPHISM

- “Polymorph” = “many forms”
- Polymorphism is used when we need type-specific behavior from our objects. The behavior of a given object varies depending on its type.
- Polymorphism is best understood from the side of the client, so let’s take an example: Battle strategies in a real-time strategy (RTS) game.


Triarii
Wealth, experience and age make these battle-hardened veterans a fearsome force.
148 (160) 0 XP:
Weapon Damage 35
Melee Attack 48
Armour 115
Health 55
Base Morale 73
When a unit spots an enemy unit, it may either
Engage
Defend
Run away

POLYMORPHISM – RTS GAME

- Implementation using a *polymorph* strategy



AARHUS
UNIVERSITY