BTH001
Object Oriented Programming
Lesson 03
Introduction to Inheritance

Inheritance - when?

If you realize that two or more classes have common properties and/or behaviour.

Example:

Student:	name, email, start year,…
	change name, get email, get start year …
Employee:	name, email, salary,…
	change name, get email, change salary…

Inheritance - common

If you realize that two or more classes have common properties and/or behaviour.

Example

Student: name, email, start year,…
change name, get email, get start year …

Employee: name, email, salary,…
change name, get email, change salary…

Class containing common

Create a class for the common parts:

Example

Person: name, email
change name, get email

Inheritance - specific

The classes Student and Employee also have specific properties and behaviour.

Example:

Student: name, email, start year,…
change name, get email, get start year …

Employee: name, email, salary,…
change name, get email, change salary…

Classes containing specifics

Create two classes for the specific parts :

Example

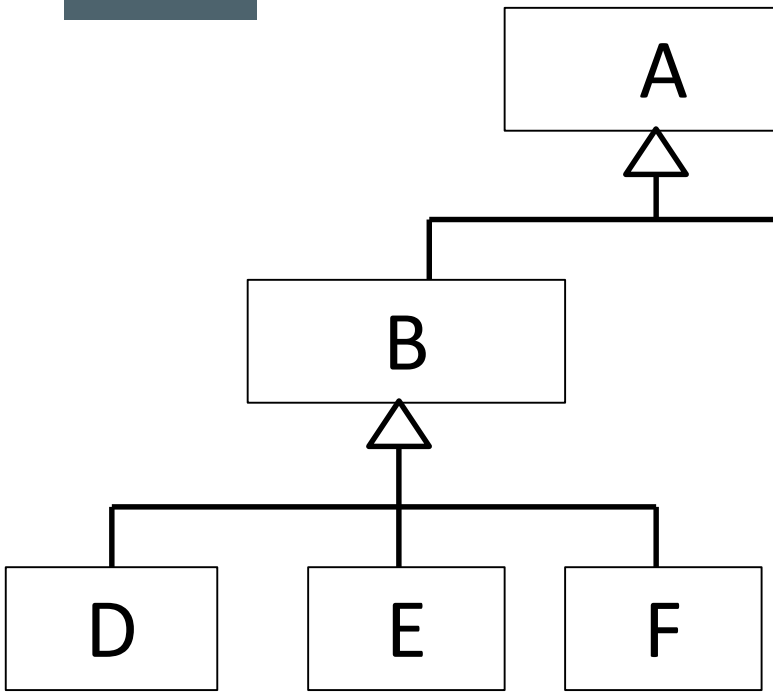
Student: start year
get start year

Employee: salary
change salary

Inheritance – when?

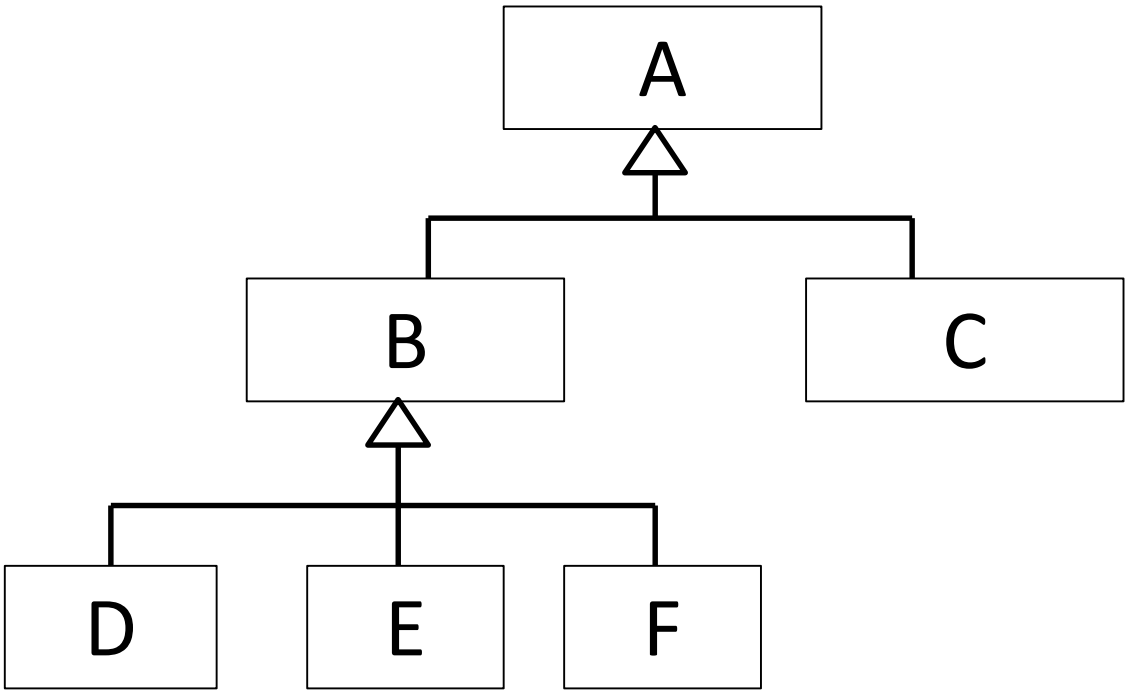
- Now we have 3 classes
  - Person
  - Student
  - Employee
- We want to use the class Person, containing the common parts, and
  - just add specific content to the Student class
  - just add specific content to the Employee class
- Use Inheritance!!!!

Inheritance – base and sub classes



A is the base class of B and C
B is the base class of D, E and F
B and C are sub classes of A
D, E and F are sub classes of B

Inheritance continued



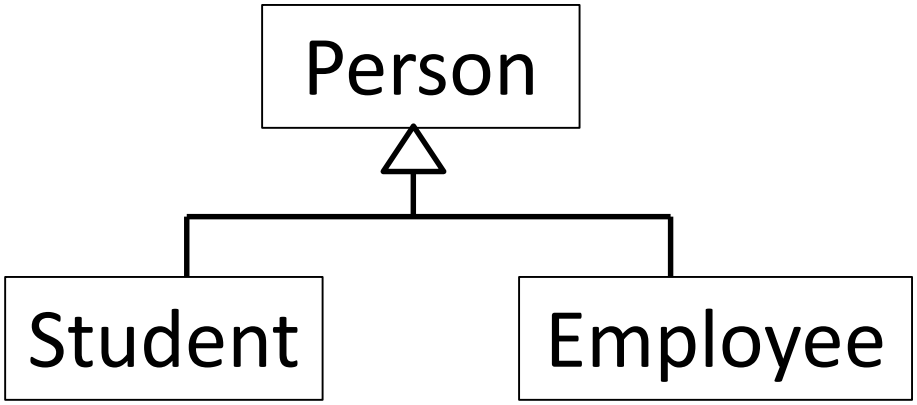
B inherits from A
D inherits from B
B is derived from A
D is derived from B

Inheritance – “Is a/an”

The relation is read ”is a” or ”is an”

An Employee is a Person

A Student is a Person



Generalization and Specialization

- Things (properties and behavior) that are common for both Students and Employees are found in the Person class
- Things (properties and behavior) that are specific for Students are found in the Student class
- Things (properties and behavior) that are specific for Employees are found in the Employee class
- Both Student and Employee inherits from Person

Inheritance in C++

```
class SubClass: public BaseClass
{…}
```

```
class Person
{
private:
   string email;
   …
public:
   string getEmail();
   …
};
```

```
class Employee: public Person
{
private:
   int salary;
   …
public:
   void setSalary(int salary);
   …
};
```

What is inherited?

- All members are inherited, though with different access
- Public members in the base class are accessible from the subclass
- Private members in base class are not accessible from the subclass

Person - Employee

- An Employee object ”contains” a **name** and a **salary**
- An Employee object can perform **getName()** and **getSalary()**
- From the Employee class it is not possible to access the member **name** - it is private in the Person class
- From the Employee class it is however possible to access the function **getName() -** it is public in the Person class

Another example

A first draft of a Game in which a player has to avoid 4 enemies falling from the top of the window.

When the player is hit by an enemy the player looses health and the enemy disappears. The health that the player looses depends on the damage the enemy causes.

Another example (continued)

Both player and enemy has a texture, a sprite and a speed. Both player and enemy is drawable.

A player also has health and an enemy instead has damage.

A player moves according to specific keys that has been pressed. A player can loose health and a player can deliver its health.

An enemy only moves downwards. An enemy can deliver its damage and an enemy can disappear.