

Exercise Class 13

Contents

1	Inheritance and Polymorphism - <i>Blackboard</i>	1
2	Modeling Using Inheritance* - <i>Blackboard OR Programming Environment</i>	1
3	Recap	4

1 Inheritance and Polymorphism - *Blackboard*

At this point you can explain the students inheritance and polymorphism (virtual member functions and base classes).

You can think of an example of your liking (e.g. Animal->Mammal->Bird) and the implement a virtual member function for that. Make sure to include some data members in the base class and sub class). In exercise 13.2 the students are supposed to use shape as base class for more specific classes and implement a virtual member function. Try to prepare for that.

Questions?

2 Modeling Using Inheritance* - *Blackboard OR Programming Environment*

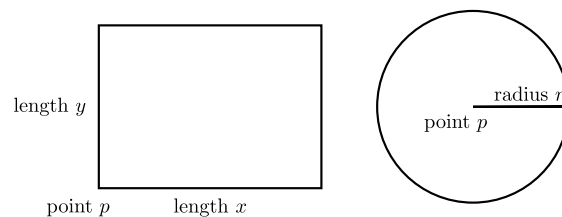
This section is voluntary. It might help your students to understand the concept of inheritance.

The following examples are supposed to highlight the difference between inheriting from a class and having an instance of the same class as private data member.

Let's start by defining a class for points in two dimensions.

```
class point {  
private:  
    double x;  
    double y;  
};
```

Now imagine we wanted to write a `class` rectangle and a `class` circle, which have a point object as the lower left corner or as center respectively.



Then a rectangle needs 2 additional numbers, the lengths of the left and the lower edge, to be fully defined, and a circle needs one more number, the radius, to be fully defined. Now we have at least the following two possibilities to implement this structure. The first one uses inheritance:

```
class rectangle: public point {  
private:  
    double lengthx;  
    double lengthy;  
};  
  
class circle: public point {  
private:  
    double radius;  
};
```

And the second one just adds a point object to the private data members of the rectangle and circle classes.

```
class rectangle {
private:
    point p;
    double lengthx;
    double lengthy;
};
```

```
class circle {
private:
    point p;
    double radius;
};
```

Both can be done and both are viable. Is there a reason why we should prefer one of them? One language crutch that can be used to distinguish between inheritance and private data members is the question “IS a circle a point or does a circle HAVE a point?”. It being a point suggests inheritance, it having a point suggests adding that point as private data member. For abstract objects like points and circles this might not work so well, a circle “having a point” might sound a bit better, but all in all this is not very satisfying. This language crutch applies better when we model things which are closer to real life and hence closer to everyday language.

Imagine three classes with the names Student, Legi, Phys_Student. Now ask the questions again: “Is a student a legi or does a student have a legi?” “Is a phys-student a student or does a phys-student have a student?” By asking these questions put the following classes in relation to each other: Student, Legi, Phys_Student, Math_Student, University.

Here is an example solution (*Note: the classes below are ordered “top down” with respect to logic. If you would implement the classes you would have to further pay attention to scopes.*):

```
class University {
private:
    std::vector<Student> students_;
};

class Student {
private:
    Legi legi_;
};

class Phys_Student: public Student {
```

```

private:
    int semester_;
};

class Math_Student: public Student {
private:
    int semester_;
};

class Legi {
private:
    int immatriculation_year_;
};

```

It is important that there is no “one right way” to structure these class relations. Using language crutches like in the above example just helps us getting started with our model. But during the geometry example we saw that language sometimes is not much of a help either.

Questions?

3 Recap

Discuss topics where your students had trouble. Discuss topics you think that are important for your students. Discuss old exam questions. ...
The goal of this section is to give your students a good preparation for the mock exam.

Questions?