Principles and Applications of Programming in C++ Lecture 7: Encapsulation

Dr Matthew Yee-King

Goldsmiths, University of London

November 22, 2017

Lab review

- 3d program

New topic: Object oriented programming

- What are classes and objects?
- Class members
- Interfaces
- Lifecycles
- Principles of good class design

Lab review

-

Quiz review

Look at the statistics for the quiz.

What does Bjarne say about classes?

> *[p213] A class is a (user-defined) type that specifies how objects of its type are represented, how those objects can be created, how they are used, and how they can be destroyed.*

What does that tell us?

Activity: predict things you'd find in a class

[p213] A class is a (user-defined) type that specifies how objects of its type are **represented***, how those objects can be* **created***, how they are* **used***, and how they can be* **destroyed***.*

Q1: Class X

```
class X {
public:
    int m;
    int mf(int v);
};
```

Classes have members

- Data members define data that is stored in each instance of this class (each object).
- Function members define behaviours that objects of this class have.

Think about how you would model a character in a video game

1. State (data members)
2. Behaviour (function members)

Interface

The interface is what we can see from outside the class.

Q2: What do you think we can see from outside this class?

```
class Car {
  public:
    Car(int _cylinders);
    void rev();
    void getCylinders();
  private:
    int cylinders;
    int speed;
};
```

Objects have a lifecycle

1. Construction
2. Activity
3. Destruction

Think about what happens in the video game character's lifecycle

1. What goes in the constructor?
2. What are the activities?
3. What do you think goes in the destructor? when does it run?

Constructor syntax

```
class Car {
  public:
    Car(int _cylinders);
  private:
    int cylinders;
    int speed;
};
```

```
Car::Car(int _cylinders)
  :cylinders{_cylinders},speed(0)
{
     // do non-initialisation stuff here
}
```

Worked demonstration of object lifecycles

Lets see when the following things happen to a 'car' object.

1. Construction
2. Activity
3. Destruction

Bjarne's principles of good class design

- **Keep interfaces complete.**
- **Keep interfaces minimal.**
- **Provide constructors.**
- Support copying (or prohibit it)
- Use types to provide good argument checking.
- Identify nonmodifying member functions
- Free all resources in the destructor

(p225 in the Programming: Principles and Practice Using C++)

Example: table class

Now let's try to refactor some example code that draws a table into a class.

Arrays of objects

You can create arrays of objects.

Pick a standard function and say what it does

Activity

Q5: What is ofPixels for?

Quiz