CS7025 Programming for Digital Media

Lesson 7 – OOP

Recap

JavaScript Objects

```
let student = {
    name: "John Doe",
    module: "CS7025",
    grade: calculateResults(68, 52, 82, 66)
}
```

JSON // all keys need to be surrounded by quotes, no functions

Object Oriented Programming

Object Oriented Programming
In JavaScript

Principles of OOP

▶ Abstraction   // blueprint

▶ Inheritance   // sub classes

▶ Encapsulation   // private members/methods

▶ Polymorphism   // different behaviour same name

▶ Method Overloading / Overriding // different method depending on
                                            // number of arguments

Object Oriented Programming
In JavaScript

In OOP you use classes to create a (reusable) blueprint of an object with the data and the methods it needs

```
class Person{
    name;
    constructor(name){
        this.name = name;
    }
    speak(){
        console.log(`Hello, my name is ${this.name}`);
    }
}
```

Object Oriented Programming
In JavaScript

When you create an instance of a class, you create an object with the characteristics of that class, all its members (variables and methods) are available.

```
let joris = new Person("Joris");

joris.speak();
```

Object Oriented Programming
In JavaScript

Inheritance

You can extend an existing class to give it more specificity, new members or change existing ones

```
class Fish extends Animal{

    move(){

        console.log("swim");

    }

}

let guppy = new Fish();

guppy.move(); // "swim"
```

Object Oriented Programming
In JavaScript

Encapsulation

You can control who has access to members by making them private

```
class Person{
    #name;
    constructor(name){
        this.#name = name;
    }


    getName(){
        return this.#name;
    }
}
```

```
let jim = new Person("Jim");


jim.name        // undefined
jim.#name       // Error
jim.getName() // "Jim"
```

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Object Oriented Programming
In JavaScript

Polymorphism

```javascript
class Animal{

    name;
    constructor(name){

        this.name = name;

    }

    move(){

        console.log("moving!");

    }

}
```

```javascript
class Bird extends Animal{

    move(){
        console.log("Fly");
    }

}



class Fish extends Animal{
    move(){
        console.log("Swim");
    }
}
```

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Object Oriented Programming
In JavaScript

Static Class Members

Sometimes you just want to access a class member without instantiating the class. `Math.random()`

```
class Car{
    static drive(){
        console.log("vroom, vroom");
    }
}


Car.drive(); // "vroom, vroom"
```

Try it yourself

Thank You

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin