

Lecture 5: Object Orientation

Curtin FIRST Robotics Club (FRC) Pre-season Training

Scott Day

265815F@curtin.edu.au

November 25, 2016

Curtin University

Table of contents

1. Object Orientation
2. Classes and Objects
3. Inheritance
4. Overloading
5. Polymorphism
6. Abstraction
7. Encapsulation
8. Interfaces

Object Orientation

Core of Object Oriented Programming (OOP) is to create objects, in code, that have certain properties and methods.

While designing C++ modules, we try to see the whole world in the form of objects.

For example, a car is an object which has certain properties such as color, number of doors, and the like. It also has certain methods such as accelerate, brake, and so on.

Lecture 5: Object Orientation

└ Object Orientation

└ Object Orientation

One of Object Oriented Programming (OOP) is to create objects, in code, that have certain properties and methods.

While designing C++ modules, we try to see the whole world in the form of objects.

For example, a car is an object which has certain properties such as color, number of doors, and the like. It also has certain methods such as accelerate, brake, and so on.

Prime purpose of C++ was to add object orientation to C, which is in itself one of the most powerful programming languages.

There are a few principle concepts that form the foundation of OOP:

Object The basic unit of OOP. Both data and function that operate on data are bundled as a unit called an **Object**.

Class The blueprint for an object.

Abstraction refers to, providing only essential information to the outside world and hiding their background details.

Encapsulation is placing data and functions that work on that data in the same place.

Inheritance is the process off forming a new class from an existing class that is from the existing class called a base class. The new class formed is called the derived class.

Polymorphism is the ability to use a function in different ways.

Overloading is also a branch of polymorphism. It allows you to specify more than one definition for a **function name**.

Lecture 5: Object Orientation

Object Orientation

Overview

Overview

There are a few principle concepts that form the foundation of OOP:

Object: The basic unit of OOP. Both data and function that operate on data are bundled as a unit called an **Object**.

Class: The blueprint for an object.

Abstraction: refers to, providing only essential information to the outside world and hiding their background details.

Encapsulation: is placing data and functions that work on that data in the same place.

Inheritance: is the process of forming a new class from an existing class that is from the existing class called a base class. The new class formed is called the derived class.

Polymorphism: is the ability to use a function in different ways.

Overloading: is also a branch of polymorphism. It allows you to specify more than one definition for a function name.

- **Class** This doesn't define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object.
- **Abstraction** I.e. to represent the needed information in program without presenting the details.
- E.g. A database system hides certain details of how data is stored and created and maintained. C++ classes provides different methods to the outside world without giving internal detail about those methods and data.
- **Inheritance** Most useful aspects of OOP is code reusability.
- Very important concept of OOP since this feature helps reduce the code size.
- **Polymorphism** Poly refers to many.
- A single function or an operator function in many ways different upon the usage is called polymorphism.

t

Classes and Objects

t

t

t

Inheritance

t

t

t

Overloading

t

t

t

Polymorphism

t

t

t

Abstraction

t

t

t

Encapsulation

t

t

t

Interfaces

t

t

t

