

# OBJECT-ORIENTATION PRINCIPLES

DR IAN CORNELIUS

# HELLO

- Learning Objectives
  1. Understand the concept of object-oriented programming
  2. Demonstrate your knowledge of object-oriented programming

# INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING (OOP)

- Object-orientation is a software engineering concept
  - based upon the concept of classes and objects
  - they are used for modelling the real world entities
- An object-oriented program will consist of a group of cooperating objects
- Objects can create other objects and send messages to each other
  - i.e. calling each other's functions/methods
- Each object will belong to a class, where the class defines properties of its objects
  - the data type of the created object will be its class

# ADVANTAGES AND DISADVANTAGES OF OOP

## ADVANTAGES

- **Re-usability:** re-use the same source-code instead of re-writing it time and time again
- **Easier Troubleshooting:** able to locate the parts of your code that are not working correctly quickly
- **Productivity:** a larger number of libraries are readily available
- **Problem Solving:** more complex problems can be broken down into manageable chunks

## DISADVANTAGES

- **Learning and Adapting:** it does not come naturally to think as an object
- **Application Size:** more lines of code results in a larger code base
- **Applicability:** object-oriented programming is not suitable for all applications

## COMPONENTS OF OOP

- Important components of object-oriented programming are: classes and objects

### CLASS

- A group of objects that have common properties
- Considered to be a template or blueprint from which an object is created
- Classes are a **non-primitive** data type, meaning that they are user-defined
- The members of a class are access modifiers, objects, methods, instance variables and constructors

### OBJECTS

- An object is an instance of a class
- Any entity that has a property and behaviour are known to be an object

## MESSAGES IN OOP

- Messages are a request for an object to perform one of its operations
  - simply, the object is calling a function/method of its class
- Messages will be used to define an **interface** to the object
  - i.e. everything an object can do will be represented by the message interface
- The interface will often be used to provide abstractions
  - i.e. you should not know anything about the implementation in order to use it
- An interface is a set of operations (methods/functions) that a given object can use
  - i.e. `setColor()`, `getColor()` etc.

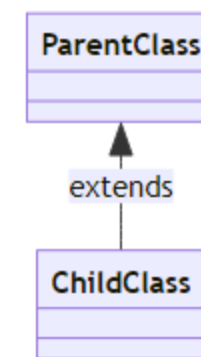
# PRINCIPLES OF OOP (1)

- OOP covers the following principles:
  1. Inheritance
  2. Polymorphism
  3. Abstraction
  4. Encapsulation

## PRINCIPLES OF OOP (2)

### INHERITANCE

- A mechanism where one object will acquire the properties and behaviours of a parent object
  - otherwise, known as a **parent-child** relationship
- A class can extend another class, hereby inheriting all of its data members and methods
  - known as children
  - the child class can redefine some parents class' members and methods and/or add its own
- A class can implement an interface, implementing all the specified methods





## PRINCIPLES OF OOP (3)

### POLYMORPHISM

- Polymorphism is the ability to take more than one form
  - i.e. a class can be used through its parent class interface, but the subclass may override the implementation of an operation it has inherited
- This will allow abstract operations to be defined and used
  - abstract operations will be defined in the parent class's interface and implemented in the child class

## PRINCIPLES OF OOP (4)

### ABSTRACTION

- Abstraction means ignoring the irrelevant features, properties or functions and providing emphasis on the relevant ones
  - providing features that are only relevant to the given project
- Abstraction is a process of managing complexity
- We abstract daily in the real-world, by looking at an object, we only see those things that have meaning to us
  - we then abstract the properties of the object and keep only what we need
- It allows us to represent the complex reality in a simplified model
- Abstraction can be used to highlight the properties of an entity that we are most interested in and hide the others

## PRINCIPLES OF OOP (5)

### ENCAPSULATION

- Encapsulation is where all the data members (or variables) of a class will be declared private
  - this could also include some methods/functions
- Classes interact with other classes only through the class constructors and public methods
- Constructors and public methods of a class serve as an interface to the class

# GOODBYE

- Questions?
  - Post them in the **Community Page** on Aula
- Contact Details:
  - Dr Ian Cornelius, [ab6459@coventry.ac.uk](mailto:ab6459@coventry.ac.uk)