Focus of this course is practical examples.

What do we want?

Code that’s maintainable.

Code that flexes when changed and doesn’t break.

Code that’s extensible. Can be easily extended to meet new requirements

Code that is easily testable.

Interfaces help us get there.

What are interfaces?

Defines a contract consisting of a collection of public members. When a class implements an interface is agrees to a contract and must implement each member.

NotImplementedException. An exception used in concrete parent class methods to force overriding in the child class.

An interface can be described as a purely abstract class. None of the members are implemented.

How do we decide between an abstract class a concrete class and an interface?

The concrete class doesn’t force us to override methods which means the errors appear at run-time.

Abstract class and Interface force us to provide an implementation.

Use an abstract class if it can contain implementation code that can be shared amongst all the child classes.

An interfaces biggest strength is a class can implement more than one interface.

In c# there’s only single inheritance but can implement any number of interfaces. It can fulfil many roles.