Abstraction: Basically it is the technique of mapping real life concepts into a program or code as objects. This helps us define an object class and its roles and ensures that it only shows users what they need to know and hide unnecessary details.

For instance, we have all seen books in library and saw that they have a name, author name and ISBN no. They can be loaned from the library as well!

Thus in the code for book class, there are fields for name, author, isbn, onloan. The book constructor has only name, creator (ie author name), and isbn in it as these 3 things make up the book! Then there is the readonly property for name, creator and isbn as these have been set from before and we (the readers) can't change it! Also it has Onloan method which tells us whether the library has that book now or if it has been borrowed and not returned!

(here I had spoken about the book class before the modification via library resource!)

Polymorphism: polymorphism mainly means "of many forms". Here we are using it to say that the parent class can provide placeholder methods and the child class can decide whether to use that method or override it to.

For instance, here (after the modification) the library resource is the abstract parent class of the child classes book and game. Since the child classes have the exact same readonly property name and property onload, these can be written in the parent class only and the child classes can just inherit from it. But the child classes have differences in the readonly class creator as book is using it to get author while game is using it to get the developer. Thus it can be kept in parent class as a virtual method while the child classes can override it to suit their needs.

Reference: Lecture notes from week 1 and 4