Answer to 1)
Inheritance is a concept of the oriented-object programming paradigm.
It allows one class that inherits from another to get the same members. Members are properties, methods and fields.
One class that inherits from another is called a derived class or a sub-class.
I would use it to create a class slightly different.
For example, I have a Publication class. I would use inheritance to create Book or Newspaper classes.

Answer to 2)
Polymorphism is another concept of oriented-object programming.
It enables one class to be derived into many other classes with different forms from the same base structure.
One class can for example change a method of the base class by overriding it.
In addition, some classes are meant to change shape, such as the abstract classes or the interfaces.

Answer to 3)
One advantage of the static method is that they can be called through the name of the class without needing to be instantiated.
I would use a static property when I need it outside of my class.

Answer to 4)
An interface is a class in C#. Any class that inherits from an interface MUST implement its members.
Nevertheless, the methods have to be left empty inside of the interface.
It is a useful concept when one wants to force a subclass to implement methods.
I would use an interface to define a Shape class with an area, then create classes like cube, sphere, square that inherit from it.

Answer to 5)
Abstract classes are much like the interfaces, except that only the abstract members need to be re-implemented in the sub-classes.
It is a powerful concept to create small variations of classes.
I could use the same example as before.