Main Ideas

Objects and closures seem closely related
Closure is an important concept in Functional Programming
SOLID principles lead to a style of design that makes Functional Programming quite attractive.

SOLID

Single Responsibility (SRP)
Open/closed (OCP)
Liskov substitution (LSP)
Interface segregation (ISP)
Dependency inversion (DIP)

SOLID (ext)

SRP	a class should have only a single responsibility
OCP	“software entities … should be open for extension, but closed for modification.”
LSP	“objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program.”
ISP	“many client-specific interfaces are better than one general-purpose interface.”
DIP	one should “depend upon abstractions, [not] concretions.”[

Object (Data with behaviour)

```
public class FileStore : IMessageQuery
{
    private readonly DirectoryInfo workingDirectory;
    public FileStore(DirectoryInfo workingDirectory)
    {
        this.workingDirectory = workingDirectory;
    }
    public string Read(int id)
    {
        var path = Path.Combine(
            this.workingDirectory.FullName,
            id + ".txt");
        return File.ReadAllText(path);
    }
}
```

Closure (Behaviour with data)

```
var workingDirectory = new DirectoryInfo(Environment.CurrentDirectory);
Func<int, string> read = id =>
    {
        var path = Path.Combine(workingDirectory.FullName, id + ".txt");
        return File.ReadAllText(path);
    };
```