10-3

OOP’s

Abstraction

* Find common characteristic, generalize
  + Accomplished with classes and interfaces
* Separation between needed functionality & implementation details

Encapsulation

* We are able to treat related data/behavior as a single unit
  + Objects, access modifiers

Polymorphism

* Ability to substitute different implementations
  + Method overriding, method overloading, generics(list class), delegates

Inheritance

Serialization – arrange it in a line/order in order to

* Write to storage
* Send over network

Deserialization – interpreting some kind of text or binary format and desterilizes it back into objects

Formats

* Your own binary format, to save space
* Custom text(XML, JSON)
  + XmlSerializer
  + JSON – DataContractSerializer
  + JSON.NET

Enum

Serialization

Async

* Asynchronous programming – Simple console app runs on only one thread
* Why do we do this? Allowing slow network to go while other program is running??
* Syntax? Await, async, tasks
* Void->task & T ->Task<T>
* WHAT IS T?
* Switching sync to async
  1. Call the async version of whatever method is going to access network/other slow thing
  2. Await the task returned by that method
  3. Use modifier async on your method
  4. Never do an async void, you make your method return a task
  5. Add “Async” suffix to the name of your method
  6. Repeat from step 1 on up to any callers of your method

Delegate

* What is it? – it can be any method with a certain return type and parameter types
* Delegate Types – (like struct and enum’s classes,events)
* Generics func and action replaces delegates

Events

* Entities that work on a publish subscribe idea
* C# lets you subscribe event handler delegates to events.

Generics

* Allows you to write code/use library methods which are type-safe, i.e. a List<string> is guaranteed to be a list of strings.
* are you trying to put an int into that list of strings? Using an ArrayList would cause that to be a less transparent runtime error.
* Faster than using objects as it either avoids boxing/unboxing

Void - > Action

Anything Else -> Func

LINQ

* Language integrated query
* 2 syntaxes(method and query)
* Nothing in LINQ modifies the original collection or any elements in it always produces a new collection
* LINQ Methods
  1. Those that return single values(execute right away) ex. First(pass a condition and returns first in list that matches),Sum, Average, Single…
  2. Those that return IEnumerable (some sequence) those use “deferred execution”(do only processing when you need an actual result – doesn’t execute the operations until you extract an actual value
  3. Stuff like ToList, ToArray those force the execution right there rather than deferred ex. Select, Where, Take Skip, GroupBy
* Can also be converted to sql and other kinds of data sources

Extension method?

* this keyword
* what is it and
* why is it important/useful?