10/4/2019

Variance (As, Boxing, Casting, Is, Out, Ref, TypeOf)

Casting/Type Conversion

* Among numeric types, conversions that could lose any data must be explicit with casting operator () otherwise they can be implicit
  + Int five =5;
  + Double otherFive = five;
  + Int nextFive = (int)otherFive;
* Conversions when type heirarchies are concerned
  + Var list = new List<int>();
  + Object o = list; ->implicit upcasting
  + List<int> listAgain = (List<int>)o; ->explicit downcasting

Generic Constraints

* What parameters are allowed(class->reference(so int can’t be passed and new->????)

Try parse & parse = out parameters

Ref params

* are like out parameters, but more flexible
* closest thing sort of pointers to pointers
* if you call a method youre making a new variable(they can’t escape) unless your’e doing ref
  + static void Swap<T>(ref T one, ref T two)
  + {
  + T.swap = one;
  + One = two;
  + Two = swap;
  + }

Type Parameters