**Name: Patrick O’Brien**

For each of the below questions, write a short sentence or two to express (in your own words) your answer. Keep the answers short, but use complete, correct, English sentences.

If it helps to clarify the questions, feel free to mentally prefix all the questions with the phrase "According to the video…"

1. After you’ve watched all the videos, please answer this question:  
   Of all the videos that you watched, if you could pick one video to be re-recorded by the instructor outside of class which would you choose? Why?  
   (Keep in mind the recording outside of class will omit any pauses from the instructor answering student questions, have less hemming and hawing, etc, and generally be more concise)

|  |
| --- |
| Improve the explanation for step over and step into. |

**VIDEO: Debugging, Part 1**

1. Other than staring at the source code a lot, what other way can people debug their code (without using the debugger feature built into something like Visual Studio)?

|  |
| --- |
| Using print statements to figure out where the program starts doing something wrong. |

1. What is good about the approach you listed in the prior question? What is bad about it?

|  |
| --- |
| It can be good for checking for different ouputs, for instance, if you are creating a log file.  Downside: You have to manually add all the print statements, recompile the program, run everything again. If not, go back and add some more print statements. Slow and labor intensive. |

1. What does a “breakpoint” do?

|  |
| --- |
| A breakpoint stops the debugger at a specific line, so that you can see what is happening at that point in the execution. You can check to see if the program runs okay up to the breakpoint. |

1. Describe (at least) 2 different ways that you can toggle a breakpoint for a given line:

|  |
| --- |
| Toggle breakpoint (f9)  Can go over to the margin and click the dot to place the red dot. |

1. In Visual Studio, how do you start running the program so that the debugger can be used to debug your program?

|  |
| --- |
| Use the “Start with debugging” option when executing the program. |

1. When the running program hits a breakpoint, does the debugger stop before or after the execution of the line that the breakpoint is on?

|  |
| --- |
| It stops before the execution of the line the breakpoint is on. |

1. How can you examine the value of a variable using your mouse?

|  |
| --- |
| You select the line and a popup option is given. If you select the popup option, it will give you information regarding the value of the variable. |

1. What are the two windows that you can use to examine variables? What does each one do (and how are they different)?

|  |
| --- |
| Debug-windows-locals -> puts things that it thinks you want to look at  Otherwise you have a window that lets you specifically search for different variables. |

1. What information does the “Call Stack” window display? How is this useful?

|  |
| --- |
| Call stack displays how you got to a place. |

1. What can you do with the “Immediate Window”?

|  |
| --- |
| It’s an interpreter that lets you enter code as if it were the line you have selected. It’s useful for checking different functions. |

1. What does the “Step Into” menu option do? What does the “Step Over” menu option do? What does the “Step Out” menu option do?

|  |
| --- |
| Step into= steps into any method calls on that line. Because no constructor it will act as step over.  Step over= executes current line, but is not forced to do it. |

1. If the program is currently executing a method (like getX() ) and you want to examine the state of the **main** function, how can you use the Call Stack to do that?

|  |
| --- |
| You double click the word for main in the call stack.  After the double click the call stack it will bring you to the main function. Mouse over the function at examine the current value/state. |

(The video ends abruptly because the instructor pressed the F10 key, which tells Visual Studio to Step Over and also (accidentally) tells the recording software to stop recording – D’oh!!! ☺ )

**VIDEO: Debugging, Part 2**

1. What is another name for the printing-out-information approach?

|  |
| --- |
| Printf debugging |

1. What is ‘logging’, and why is it useful? Make sure to give a specific type of programming that commonly uses logging.

|  |
| --- |
| It’s particularly useful for web applications. You can attach the debugger to a running webserver, set a breakpoint in a web app. Web programming |

**VIDEO: Enums**

1. What are enumerations generally useful for? What will we typically be using them for in this class?

|  |
| --- |
| Turn values, status codes, stacks and queues, collections of stuff.  Enumerate all possible status codes that may result from the collection of stuff.  Enum is a new type  enum ErrorCode  {  OK,  Overflow,  } |

1. Copy the ErrorCode example from the video here, and make sure that you’re clear on the syntax (how you need to type this stuff). Make sure to fix any Word auto-corrections that happen.

|  |
| --- |
| ErrorCode ec;  ec = ErrorCode.OK;  if (ec == ErrorCode.OK)  {  // things went well…  } |

1. What is the goal of the ErrorCode enum, specifically using the example of the SmartArray.Add() method?

|  |
| --- |
| The goal is to give feedback when something happens with the SmartArray.  For instance, whenever something is added to the smart array, the ErrorCode.OK notifies the user that something was added to the smart array OK.  Then you can check to see if the errorcode is equal to preferred errorcode. If it is, then you can make the code do something specific. |

1. Given a variable named **ec** (of type ErrorCode), how could you check if it has the OK value? Give a short snippet of C# code in order to illustrate how to do this:

|  |
| --- |
| ErrorCode ec;  Ec = ErrorCode.OK;  If (ec == ErrorCode.OK)  {  // Things went well  } |

1. Given a variable named ec (of type ErrorCode), how could you check if it does **NOT** have the OK value? Give a short snippet of C# code in order to illustrate how to do this:

|  |
| --- |
| ErrorCode ec;  ec = ErrorCode.OK  if (ec != ErrorCode.OK)  {  // things did NOT go well  } |

1. How will we be using ErrorCode enums in this class?

|  |
| --- |
| Using enums extensively in this class to:  Public ErrorCode StoreData(int data)  {  // if unable to store it for whatever reason then…  Return ErrorCode.Overflow;  }  Previously using Boolean, you could only say something happened, not what it was.  With enum it allows you specify exactly what happened. |

1. When you declare something to be an enum, what data type does the compiler translate that into?

|  |
| --- |
| The compiler first translates an enum into:  An integer type |

1. How can you specify the value you want an element of your enumeration to have (for example, how could you specify that OK should have the value 0)?

|  |
| --- |
| enums are a C# value type.  You can specify by stating that OK = 0, in an enum declaration.  Otherwise:  public ErrorCode StoreData(int data)  {  return (ErrorCode)100; // bogus value  } |

1. For all the elements of the enum that you do NOT explicitly assign a value to, how does C# figure out what value to use?

|  |
| --- |
| If it is unspecified, it will take the value of the previous thing in the list and add 1 to it. So if the prior value is 200, then the next value will be 201. |

**VIDEO: Exception Handling: Overview**

1. What is major objective of exception handling?

|  |
| --- |
| Centralizing all your error handling so instead of having to individually handle every event. It removes the constant interruption of handle errors when running through the logic of your program.  With exception handling it can be at the end of the function. |

1. What is the major downside of using the older model of error handling?

|  |
| --- |
| It dilutes the flow of your program. You are forced to individually run each method and check for errors. |

1. What is the major advantage of exception handling (error-handling-wise)?

|  |
| --- |
| You get centralized error handling – the rest of the code focuses on the core logic  You can through your own exceptions, tailored to specific cases  You can catch in other methods  The schema is great for constructors |

1. Should you use Exception Handling to handle normal, expected situations in your code?   
   (Give one example of a normal situation that’s not an exceptional situation, and therefore one that you should NOT use exception handling for)

|  |
| --- |
| No, you should not use exception handling to handle normal, expected situations in your code.  Running past the normal execution length and throwing an error. |

1. Write out a quick example of C# source code that demonstrates the try, catch, and finally blocks.  
   (You can leave empty the stuff between the { and } – I want to mostly make sure that you saw the red text that highlights the syntax of exception handling)

|  |
| --- |
| Try  {  TemperatureRecord[] recs = helper.ReadFile(“TestFile.txt”);  Console.WriteLine(“Found the following {0} records”, recs.Length);  For (int I = 0; I < recs.Length; i++)  {  recs[i].Print();  }  catch (Exception e)  {  Console.WriteLine(e.Message);  }  finally  {  // Happens regardless of exception  } |

1. What happens if an exception happens inside a try block?

|  |
| --- |
| It will run the specified exception in the catch block. If the catch block does not specify an exception, it will run regardless. The finally {} block will run regardless of whether there was an exception or not. |

1. What is the finally block useful for?

|  |
| --- |
| Closing files, closing a connection to a database, notify the external world that you are no longer using a resource (within the try block). |

**VIDEO: Exception Handling: File I/O Overview**

1. Are you expected to be able to write File I/O code based on the material that’s being covered in this video?  
   (Note that you may cover File I/O elsewhere in this course, in which case you would then be expected to know this material well enough to write code using it)

|  |
| --- |
| Not from this specific video, but later on we will be expected to know how. |

1. What two ‘flavors’ do files come in? Briefly describe each one, and mention why we’re using a text file for this program.

|  |
| --- |
| Text files = ascii, Unicode, whatever you want. You can read and write from them.  Binary files = 1s and 0s |

1. Where will the file appear when the WriteFile method creates a new file?

|  |
| --- |
| The specified directory, or otherwise (if not specified) - the working directory. |

1. Give a quick overview of what the ReadFile method does, focusing mostly on what the method does to read the information out of the file.

|  |
| --- |
| If the line is empty – throw an exception  If the file does not exist – throw an exception  Open a file reading object  Open a text reading object within the file reading object  Call ReadLine  Feed to int32 parse command  Get first number from first line  If there are three records in the file, then each record is stored on its own line, they are separated by commas  Read the entire line out, split up into three (lat, lon, temp)  Feed into TemperatureRecord Object  Return the array with the different objects |

1. Why is there almost no error handling in the ReadFile method?

|  |
| --- |
| Because all the error handling is centralized using exception handling.  Then you only need to focus on the core logic. |

**VIDEO: Exception Handling: Try Catch in the same method that throws the exception**

1. Describe (briefly) what code is protected by the try block.

|  |
| --- |
| Protects everything down to the matching curly brace. FileIO |

1. If everything goes fine, which line will the program reach?

|  |
| --- |
| The final line to return the contents of the method. If anything goes wrong it jumps right to the catch statement and does not finish. |

1. When the third number is removed from the file, which lines causes an exception? Briefly describe the cause of the exception.

|  |
| --- |
| When the code was written, it was assumed that the file was correctly formatted.  There is no fields bracket 2 when you attempt to parse the third number. |

1. When the exception happens, where does the program’s execution jump to?

|  |
| --- |
| The catch block, which prints that the index is out of bounds and then returns null. |

1. What does the catch block do?

|  |
| --- |
| It catches the exception.  For instance in the case of this program: It writes the error message and then returns null. |

**VIDEO: Exception Handling: Try Catch in Main**

1. How do you figure out which methods throw exceptions (and which exceptions they throw)?

|  |
| --- |
| You can look them up in the C# documentation on the Microsoft website. |

1. When the program attempted to convert the string “Mike” to an Int32, what happened?

|  |
| --- |
| A FormatException when attempting to convert Mike to Int32 |

1. How does the program figure out which catch block to use?

|  |
| --- |
| It looks for the catch block that covers the exception.  If the particular block of code does not have a try catch block, it goes up the chain to check to see if whatever called that had a try catch block.  Until it finds something that catches it, or nothing that does. If nothing is found it will then crash in a controlled fashion. |

**VIDEO: Exception Handling: Throwing your own exceptions; using exception handling in a constructor**

1. What is the syntax for throwing your own exception?

|  |
| --- |
| throw new Exception(“File does not exist!”);  throw new ArgumentException(); |

1. How do you figure out which exceptions that you can throw?

|  |
| --- |
| Google it |

1. Typically, how do the different exception classes differ from each other?

|  |
| --- |
| The parameters may be different |

1. What are the catch statements based on?

|  |
| --- |
| The type of the object |

1. Explain how exception handling useful inside constructors.

|  |
| --- |
| Because constructors have no return value, you can use exceptions to give feedback. For instance if you want to know if a temperature is greater than another temp, you can  throw new ArgumentException(“….”); 🡪 also stops program |

**VIDEO: Big Oh Review**

1. What does a “profiler” do? How is this useful in the software development process? In what way / at what time is this NOT useful for software development?

|  |
| --- |
| A profiler: A tool that attaches to a program, measures how long everything takes to happen. Tells you how long each individual took to execute.  Useful: Tells you which functions could be executed faster. Good for fine tuning.  Not Useful: Telling you which algorithms to use at a general level. |

1. What level / time during the development life-cycle does the Big Oh notation help you figure out which algorithm to use?

|  |
| --- |
| The early stages of development, when you are still planning whether or not to use a certain kind of algorithm. E.g. Linear search or binary search. |

1. What is (conceptually) important about the “blocks of constant time”?

|  |
| --- |
| Regardless of whether you take the return statement or not, there is a finite amount of time that it could take you to return the value.  You can go through everything, say that at worse case, there are at most 4 different parts…  Does not depend on the size of the array, always takes the same amount of time in the worst case (upper bound).  How many times do you run the block of code. |

1. What determines how much time the linear search function will take?

|  |
| --- |
| The upper bound (worst case) for the size of the array.  How many times the block of code will be run. |

1. What is the “exact expression/equation” that’s given in the video for how much time linear search might take? What does each part of the expression mean (I.e., what does the 72 represent? The N? etc)*(Keep in mind that 72 & 113 were clearly picked arbitrarily, in order to have values – this is not necessarily an accurate expression)*

|  |
| --- |
| 72 \* N + 113  72 = number of CPU units in time  N = number of elements in the array  113 = Units of CPU time ( additional units based on how long it takes to process the extra components of the function |

1. Fill in the blank in the space provided below:  
   At the end of the day, we’re not really interested in the exact amount of time [that this particular implementation] takes, instead we’re interested in \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

|  |
| --- |
| Answering the question, is linear search better, or is binary search? |

1. O(N) is what sort of bound?

|  |
| --- |
| The upper bound or maximum amount of time for the linear search to run using the beforementioned equation. |

1. Intuitively, what will the running time be for any routine where you end up running through every single element within an array?

|  |
| --- |
| O(N) where N is the number of elements in the array.  If you have a million elements in the array, you’re going to have 1 million \* the cost of printing one thing. |

1. What will the running time be for something like the SetValueAt method?

|  |
| --- |
| It will be:  Constant amount of time (upper bound)  No matter what, you might never need more than 12 lines of code.  O(1) 🡪 Some number \* 1 which is the real number.  12 \* 1 |

1. How do you write out “constant time” **IN BIG OH NOTATION?**

|  |
| --- |
| O (1) |

1. According to the chart (within the Big Oh Review document on the website), will linear search or binary search be faster (assuming that you’re able to use either one)? Why (explain briefly and intuitively)?

|  |
| --- |
| Binary search will be faster. While linear search time increases exponentially, binary search time is logarithmic, and according to the chart, its run time is drastically lower than linear search as more indexes are added to the array. |

1. According to the table that summarizes common running times (within the Big Oh Review document on the website), what are the five most common running times?  
   (You may need to slow down / pause the video, and/or open up the document to read the five common categories)

|  |
| --- |
| O(1)  O(log\_2N)  O(N)  O(N \* log\_2N)  O(N^2) |

1. Is it normal to apply Big Oh notation to individual lines of code within a method?   
   At what “level” is Big Oh notation normally used?

|  |
| --- |
| Typically you would apply it to the entire algorithm or method. |

NOTE: There are no Viewing Quiz questions for the following videos. Instead, you should fill out the following video outlines.  
NOTE #2: There are [directions on the course website that explain how to outline the videos](http://faculty.cascadia.edu/mpanitz/Courses/BIT142/Homeworks/Outlining/index.html). These direction videos are pretty short, so please do watch them!

**Outline for “What is a modern collection class? (including "What is a 'generic' class?"), Part 1”**

**File: Collections\_Generics\_Part\_1.mp4**

* Collection class = A class whose purpose is to store a bunch of objects.
* An array is a collection of objects (technically not a class) 🡪 arrays have many limitations, difficult to find things, middle of the array you have to shove everything towards the end, hard to add new items once the array is full. Arrays are best for storing a fixed number of a specific type.
* A collection class makes it easy to store lots of objects
* Specific classes may have unique advantages 🡪 generic collection classes can be used with any one type, in a type-safe way (meaning the compiler can check that you’re only using that specific type, at the compile time).
* Examples: Class Collections: These were included in C# 1.0 Relied on inheritance to store things
  + Ex: System.Collections.ArrayList
  + System.Collections.Stack – replaced by generic
  + System.Collections.Queue – replaced by generic
* Generic Collections (preferred to use now) – They check to make sure they are using the correct type, they will also do compile time type checking.
* Ex:
  + System.Collections.Generics.List<>
  + System.Collections.Generics.Stack<>
  + System.Collections.Generics.Queue<>
  + System.Collections.Generics.Dictionary<>
* Specialty collections
  + Good to know they exist, but don’t need to know about them
* Ex:
  + Anything in System.Collections.Concurrent.\*

**Outline for “What is a modern collection class? (including "What is a 'generic' class?"), Part 2”**

**File: Collections\_Generics\_Part\_2.mp4**

* Generics
* Why they were added to C#
  + Flow control: if, while loops, for loops
  + Classes, etc 🡪 made for programmers
* Generics are made for programmers who are creating the libraries for programmers to use.
  + ‘Generics’ were added to C# to improve the collections classes
  + A ‘collection’ class stores a bunch of data
  + An array can be thought of as a collection of data
  + There’s lots of other ways of organizing data (other collections)
* We CAN use generic classes to improve our code, but you won’t use it every day.
* A class that be used to store several different types of data.
  + A normal class only works with a single type of data
  + A normal class will have to use inheritance to fulfill the same function as a generic
* Ex:
  + ArrayList() vs List<int>()
* ArrayList() accepts anything
  + Makes it easier to accidently put wrong data in
  + Also is more complicated to get data out
* List<int> will only accept ints
  + Compiler stops us from accidently putting wrong data in
  + Easier to get the data back out
  + Can be changed to take classes, or different data types
* ArrayList
  + Int num = (int)numberList[i]; checks to make sure that it is an integer being fetched.
  + DemoClass obj = new DemoClass(100) adds the entire object to the obj when adding to the   
    numberList.Add(obj) and cannot be converted to int (it’s a bug)
* List
  + List<int> genericNumberList = new List<int>();  
    genericNumberList.Add(10);
  + Not necessary to convert since compiler already knows to hold integers.
  + Makes sure the right thing happens regardless of whether it’s an int or a class, compile time benefits -- that’s why it’s called generic (takes everything).

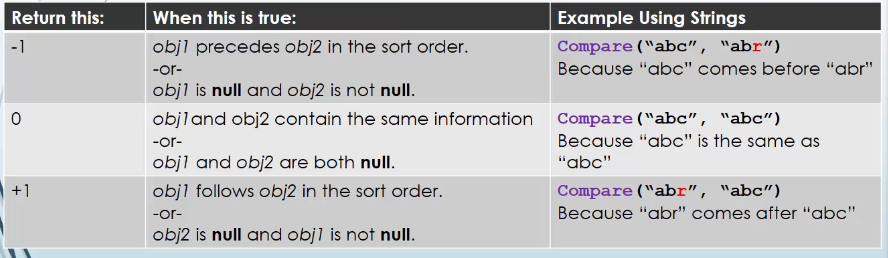
**Outline for “Creating your own generic class”**

**File: 02\_Creating\_Your\_Own\_Generic.mp4**

* Creating a generic class  
  //ex:  
  StoreInt int\_storage = new StoreInt(10);  
  int\_storage.SetData(20);  
  Console.Writeline(“int\_storage contains: {0}”, int\_storage.GetData());  
    
  //ex:  
  StoreValue<double> double\_storage = new StoreValue<double>(10.0);  
  double-storage.SetData(21.575);  
  Console.WriteLine(“double\_storage contains: {0}, double\_storage.GetData());
* // changes TypeOfData depending on the value type  
  Class StoreValue<TypeOfData>   
  {  
  private TypeOfData data;  
  public StoreValue(TypeOfData d)  
  {  
  SetData(d);  
  }
* // Constructor ex:  
  public StoreValue(TypeOfData d)  
  {  
  SetData(d);  
  }
* In the normal version,  
  public StoreInt()  
  {  
  SetData(0);  
  }  
  // list version  
  public StoreValue()  
  {  
  SetData(default(TypeOfData));  
  }  
  // Fills in the default for the selected type

**Outline for “What is an interface?”**

**File: 04\_Implementing\_IComparer.mp4**

* What is an interface
* An interface is a description of a behavior that multiple classes are required to perform.
* IComparer:
  + Generic collection like List<T>
  + Support a method to find something in the collection
    - We’re going to look at the List<> class which provides a BinarySearch<> method
* WRONG WAY // only works with a primitive
  + Public int BinarySearch<T> (T findThis)   
    {  
    if ( findThis == nextItem)  
    }
* IComparer:
  + Instead of prescribing a single way to check if two things are the same, how about we let the class decide how to compare itself.
  + Can be done by adding a method to the class
    - Eg. Public int Compare (my\_class obj1, my\_class obj2) 
  + Classes commit or promise to provide the method with a return type.
* Adding the compare method is better than == directly
* But it’s not perfect
  + If we add a compare method to a class then that class can have only one possible ordering
  + Because there’s only one compare method
* Instead of putting the compare method on the class we’ll put it on a different class.
* A small, single-purpose class who’s only use it to compare our objects.
* But how does c# know that our small, single-purpose class has a compare method?
  + Answer: unless we add something, it can’t
  + Better answer: we will use an interface to define what behavior our small class should do.
  + Look up IComparer examples and functions on Microsoft docs
* In IComparer, null comparing is allowed, considered less than any other object
* Int comparison = x.GetMake().CompareTo(y.GetMake()); = example of CompareTo with methods