[9999] FINAL EXAM STUDY GUIDE

YOU'RE ALMOST DONE!!!!!!

Here's a list of things to make sure you understand as you prepare for the Final Exam:

1. **READ MY CANVAS NOTES!**
2. **REVIEW EVERY PROGRAMMING ASSIGNMENT! Know them** backwards **and forwards.**
3. **Skim the Nutshell Book. Keep an eye out for stuff I discussed in class and in Canvas.**
4. **Make sure all of the items in Canvas' Modules area are checked off**
5. **Memorize the following key points:**

.NET and C#

- C# is used to create programs that run inside of the .NET Environment and can (optionally) interact with other .NET compatible programs easily.
- C# is a pure object-oriented programming environment where ALL data types are objects.
- Many of C#'s simpler data types are just .NET data types masquerading as basic C# types - i.e., "int" is the same as .NET's "Int32"
- C# programs are compiled into "Intermediate Language (IL)" - a low-level language which executes inside the .NET CLR (Common Language Runtime).
- Compiled C# code is stored in an "Assembly" file which typically has a name ending in "EXE" (for programs) or "DLL" (for libraries).
- The Stack is a region of memory where local variables and arguments live.
- Stack memory is automatically freed up when the corresponding variables go out of scope.
- The Heap is a (bigger) region of memory where objects live. Anything created by "new" lives on the heap.
- Periodically, the CLR's Garbage Collector discards objects that are no longer used from the Heap.
- The "using" statement lets you access classes from other class libraries in your program.
- Immutable objects cannot change after they have been created. Strings are immutable.
- Whenever you are assigning something of one type to a variable of a different type and information could be lost in the conversion, you have to include a cast.
- Use a "for" loop if you are counting the number of times thru the loop. Use a "foreach" loop if you doing something to all the items in an array/collection.
- Use a "do/while" loop if you have to go through the loop at least once, otherwise use "while/do" loop.
- In C#, all "case" sections in a "switch" statement must end with "break;"
- A method's name, return type, parameters and attributes are called its "signature."
- Method arguments can be passed "by reference" or "by value." If passed by reference, the original variable can be modified by the method it was passed to (bad). If passed by value, a copy of the variable's data is sent (good).
- By default, all arguments are passed "by value." However, remember that a variable-that-represents-an-object actually contains a pointer to the location on the heap where the actual object lives. So

when a variable-that-represents-an-object is passed into a method, the method actually gets a **copy of that reference**. That copy points to the same object on the heap as the original variable and, if the object allows it, the method can use that copy of the reference to modify the object's properties and/or call its methods. And so, effectively, if you pass a variable-that-represents-an-object to a method, you are *effectively* passing it "by reference."

- If you have several methods with the same name but different parameter lists, those methods are called "overloaded" methods.
- Optional parameters have default values that are used if those parameters are not included in the method call. Optional parameters must be placed at the end of a method's parameter list.
- Enums let you create a type with a set of fixed, named values like Colors.
- If you want to store null in a type that doesn't normally take null as a value (like an int), add a question mark to the end of the type. (e.g., "int?")

Object-Oriented Programming

- In OO Programming, an Object is a combination of Code + Data.
- A Class is a TEMPLATE for creating objects.
- Objects are instantiated (created) by using the "new" operator on a class. Memory is allocated from the heap and then the class' constructor is called so that it can initialize that memory.
- The "static" modifier causes just ONE version of that item to be created when the program runs.
- Encapsulation protects an object's internal data from arbitrary changes. It does that by using access modifiers like "private", "protected" and "public."
- Objects that just store and retrieve data are often called "Plain Old Java Objects" (POJOs). Data inside of a POJO is private. POJOs can add "Getter" and "Setter" methods to allow selective access to their data.
- C# supports .NET Properties which are essentially encapsulated fields with automatically generated Getters and Setters.
- Encapsulation, Inheritance and Polymorphism are often called "The Three Pillars of OO Programming."
- Inheritance lets you create a tree-like hierarchy of related classes. The lower classes are said to "inherit from" the upper classes.
- A subclass automatically gets ("inherits") all of the non-private member variables and methods that all of its superclasses have.
- C# does not support multiple inheritance. Subclasses can only have one superclass.
- If a method can be overridden, add the "virtual" keyword to its method signature.
- If a method in a subclass is overriding a method in a superclass, add the "override" keyword to its method signature.
- A "sealed" class cannot be used for inheritance. You can not inherit from a sealed class.
- An "abstract" class is essentially an "incomplete class" because it has one or more incomplete "abstract" methods.
- Abstract classes cannot be instantiated. Abstract classes must have a subclass inherit from them and implement their abstract methods.

- Polymorphism lets you call the same method on two different related objects. The system determines at runtime which method to actually call based on the type of the object calling the method.
- System.Object is the superclass of everything in C#.   Its methods includes: ToString(), Equals(), and MemberwiseClone() which means that everything in C# has those methods.

Advanced C# Syntax

- When a run-time error occurs, the system creates an Exception object and "throws" it to the inner-most in-scope "catch" block it can find.
- If an exception isn't caught by any catch block inside the program, the program crashes and a stack trace appears on the screen.
- The code inside of a "finally" block is always executed regardless of whether an exception occurred or not.
- Interfaces can be thought of as if they were "100% abstract" classes. (They aren't - but you can think of them that way.) All of the methods inside an interface are un-implemented.
- You can associate multiple interfaces with a class as long as that class implements all of the methods in each of those interfaces.
- Interfaces typically add a common feature or characteristic to a class. Examples include IComparable and IEnumerable.
- Generics allow programmers to specify the type that a method or collection will accept. This adds "type safety" because the compiler can check for incorrect assignments.
- Data Structures are different ways of storing related data items in memory.
- Common Data Structures include Arrays, Stacks, Queues, Lists, Linked Lists, Maps (Dictionaries), and Hashtables.
  - Arrays are very memory efficient and very quick to access. They are very slow to insert or delete from.
  - Linked Lists are the opposite: very quick for insertions and deletions, but relatively memory inefficient and slow to access.
  - Hashtables are very quick to access and very quick for insertions and deletions (usually). They are relatively memory inefficient however.
- Stacks are "Last-In-First-Out" (LIFO) and Queues are "First-In-First-Out" (FIFO) - i.e., "buffers"
- C#'s Generic Collections are flexible data structures that work with most types of C# objects.
- List is the most commonly used C# Collection.
- All C# collections implement the IEnumerable interface which means they can all be used with "foreach".
- Most C# collections implement the ICollection interface which means they support the Count property.
- Most C# data types implement IComparable and the CompareTo() method which returns -1 if the first parameter is less than the second, +1 if the first parameter is greater than the second, or zero if the two are equal.
- Lambdas are methods without names, parameter types or return types. The are mostly used as parameters to other methods.

- Anywhere you can use a Lambda, you can also use the name of an existing method.
- LINQ stands for "Language Integrated Query"
- LINQ allows C# programs to run database-like queries against data stored in memory.
- LINQ can be used in several ways including "Fluent syntax" ("blah.where().orderby().select()") and Query syntax ("from blah where i > 10 order by name select name;")
- LINQ has many utility methods that can be used in non-LINQ contexts for sorting, searching and converting to other datatypes.
- LINQ's "select" function lets you "project" (i.e. transform) the query results into different datatypes including "anonymous types" that you access via the "var" type designator.
- LINQ can also be used to access SQL data via the Linq-to-SQL and Entity Framework libraries.
- "Unsafe code" is a specially marked block of code that can use pointers. Rarely needed.

Programming Best Practices

- Always strive to write self-documenting, readable, maintainable code by choosing identifier names carefully.
- "Premature optimization is the root of all evil."
- Try to use Incremental Implementation: first get a simplified version of your program to run, then make it a little better, then see if it still runs, then repeat the process.
- Try to separate UI-specific code from the rest of your program because the UI can (and will) change.
- Methods can be used to break complex tasks down into more manageable chunks. Methods can also be used to increase code reuse.
- Well-designed methods can take multiple inputs, but should only return one output. Well-designed methods only do one "thing."
- When designing an OO program, examine a written description of the program looking for important nouns and verbs. Nouns will often become classes or fields. Verbs will often become methods in those classes.
- When using noun-verb analysis, be sure to convert passive verbs to active verbs to help determine which class the method should belong to.
- Most classes should include their own version of ToString() to help with UI and/or debugging.
- Favor immutability in your classes.
- Always test the edge cases in your code. Make sure your program can handle any input the universe can throw at it without crashing.
- POJOs should throw exceptions if invalid data is passed into them. POJOs should never have UI code,
- A constructor's job is to initialize a class' member variables - and nothing else. Constructors should be compact, efficient, and bullet-proof.
- Prefer "private" to non-private. Only make something public or protected if you are sure you need to.
- Inheritance hierarchies are difficult to update without breaking/rewriting lots of code that depends on it.
- Code that interacts with outside systems (disk files, the network, etc.) should be inside a try/catch block in case those calls fail.
- When possible, "program to interfaces, not implementations."

- Only catch exceptions that you can handle.
- If your program "opens" a resource (a file, a connection, etc.), it should do so in a using block so that the system will close the resource automatically.
- In general, Lambdas should be short. Long Lambdas don't promote code reuse and can make the code hard to read.
- A C# Attribute is metadata information about your program that is enclosed in square brackets and located above the method, class or block of code it describes. Attributes are used heavily in Unit Testing and the Entity Framework.

Applications

- Parsing is the task of breaking larger strings down into smaller items that contain the data you need.
- Useful string parsing methods include String.IndexOf(), String.SubString() and String.Split(). String.Replace() is sometimes used to "clean up" a string before String.Split() is used.
- A Log File is a line-based text file that contains information about each action a program (typically a server) takes. Information about each action is usually contained on a single, easily parsable line in the file.
- XML is a text-based format for transferring data between systems that is based on opening and closing "tags" similar to HTML
- JSON is a newer text-based format for transferring data between systems that is more compact, more readable and more flexible than XML.
- JSON.net is a popular 3rd party library that can convert objects into JSON (serialization) and JSON into Objects (deserialization).
- C# programmers can read data from the Internet using the WebClient class.
- There are four ways to sort a List collection
  - If the List contains items that implement IComparable, then you can just call List.Sort() (followed by List.Reverse() if needed).
  - If the List doesn't contain items that implement IComparable, you can:
    - Implement IComparable (if you control the class of the items)
    - Write a Comparer() method that returns -1, 0, or +1 and then call List.Sort(ComparerMethodName).
    - Call List.Sort() with a Lambda that calls CompareTo() on the two parameters
- Dictionaries can be used to count the frequency of items in a list.
- You can use a key inside of square brackets to access the value inside of a key-value storage collection (e.g. a Dictionary). For example, phoneNumbers["Chip"] = "555-1212"
- WinForms is older, simpler way to create GUI apps. Windows Presentation Foundation (WPF) is somewhat newer and better supported. Universal Windows Platform (UWP) is new for Windows 10. All three of these require Windows to work.
- MVC stands for "Model-View-Controller." MVC is common a "design pattern" where data (the Model) is managed independently from the application logic (the Controller) and the user-interface (the View).
- .NET Core is an open source port of .NET created by Microsoft that runs on Windows, Mac and Linux. It supports command line apps, web apps (ASP.NET), UWP apps and libraries. It does not

support WPF or Windows Forms apps (and UWP apps only run on Windows).

- ADO.NET provides database access for .NET programs including via the Entity Framework which automatically generates POJO-like classes from database tables.
- "VS Code" is an open source Code Editor with support for C#, Java, JavaScript, HTML and many other programming systems.