The Final Exam is *comprehensive* and will cover all material that we have gone over in class. The Final Exam will be given on **Tuesday, April 27, 2021**, via Canvas, from **8:00 AM to 10:00 AM** local Denton time. The exam will close promptly at 10:00 AM, and no additional time will be given if you are late. You will have exactly 2 hours to complete the exam. No calculators may be used on the exam.

This exam is closed book and closed notes and will be completed using the Respondus LockDown Browser with Respondus Monitor through Canvas. Therefore, you should have a microphone and working camera on your computer to take the exam. You will not be allowed to use any other resources (such as previous code you have written) that are not explicitly given here. You may *not* use or have your cell phone out. **Any violation of this would constitute cheating and result in a grade of 'F' for the class.** Also, note that taking exams in groups, sharing answers, or obtaining answers from the Internet is strictly not allowed. Please make sure you are familiar with these instructions and how to use this software *before* you start your timed exam as you will only be given 2 hours to complete this exam. *If you have any connectivity issues during the exam, please contact me IMMEDIATELY via e-mail so that it can be resolved during the exam window.* I will be continuously monitoring my e-mail during this time.

The exam will consist of a mix of 60 TRUE/FALSE and MULTIPLE CHOICE questions (plus any extra bonus questions depending on the participation level of the course evaluations) on an online exam in Canvas with Lockdown Browser. Even though you will not write code during this portion of the exam, you will be asked to evaluate code and provide the best answer based on the given options.

Some of the questions will be direct applications of the concepts you have learned, so it is vitally important to be able to understand the context of what we have been studying. You should review your *zyBooks* online textbook, the lecture notes, class code examples, as well as your lab assignments to prepare for this exam.

Please be sure to review the following topics that will appear in one form or another on the exam.

* You may be asked to analyze small segments of Python code to determine either what is output to the screen, what is the value of one or more variables after execution of the code, or similar.
* You should be familiar with the general structure of a program (i.e., input, process, output), including how the Python interpreter works (i.e., read, evaluate, print) either interactively or through a program.
  + Be familiar with how to start new blocks with the colon : and indentation.
* You should know what an algorithm and a program are and how to define them.
* You should know how to perform arithmetic calculations, including order of operations, as well as use of operator shorthand.
  + In particular, know how the floor division // and modulo % operators work.
* Be familiar with the general concept of a class, object, attributes, and methods, especially since everything in Python is an object.
  + Know the definitions of mutable and immutable, including which objects are mutable or immutable.
  + What is the difference between a function and a method?
* Be sure you are familiar with how to get input using the input() function, including type conversion to an integer or floating-point number, and display output using the print() function.
  + Be familiar with common uses with format() for formatting values such as floating-point numbers and strings (positional, inferred positional, named replacement).
* You should be familiar with the three types of errors in code including some of the common error types.
* You should know how to convert simple numbers (up to a byte) in binary to decimal.
* Be familiar with computer organization in general (see section 1.7 Computer Tour).
* Know the rules for identifiers (i.e., variable names) and be able to identify examples of valid and invalid identifiers, including keywords.
* Know how to import common modules, such as math, sys, os, random, etc. as well as a local Python module.
  + For the math module, you should be able to use the functions we have worked with, such as those on slide 27 in Chapter 2 lecture notes.
  + Be familiar with using the \_\_name\_\_ attribute in a module, whether it’s set to \_\_main\_\_ or the name of the module.
  + Know what the reload() function is and why it can be used.
* Know how to use the escape character \ as well as encode/decode values using ord() and chr().
* Be sure you are familiar with how to structure and use branching statements such as if, elif, and else statements.
  + Be able to evaluate a conditional expression.
* Be familiar with membership and identity operators.
* Be familiar with the local and global "scope" of all variables, including those defined in functions, passed as parameters, and global variables.
* You should be familiar with equality, relational, and logical operators in evaluating Boolean expressions, including operator chaining.
  + You should also know how to declare and initialize a Boolean value.
* Be familiar in general with all the container (i.e., iterable or sequence) types, such as tuples, lists, dictionaries, sets, and strings and be able to perform common operations on these various types for declaring, accessing/slicing, manipulating, adding/removing/modifying elements, etc.
  + General operations for containers such as min(), max(), sum(), len(), index(), count(), etc.
  + Know how to split or join a string.
  + Be familiar with list and dictionary comprehensions.
  + Be familiar with sort() and sorted() on lists and how they are different.
  + Know how to return keys, values, or both in dictionaries.
* Be sure you understand the syntax and how and when loops are used (e.g., while loops and for loops) to accomplish iterations for some task, including the three parts for the control variable (i.e., initialize, condition, and update).
  + How is a sentinel loop used? Counting loop?
  + Know how the range() function works and can be used in a loop.
  + Be familiar with how to access a list inside a list using a nested loop.
  + Be sure to understand what happens when a break or continue statement is used inside a loop.
  + Be familiar with the enumerate() function and how it can be used.
* Be familiar with how to work with files, including opening, reading/writing, and closing files.
  + What is the path separator and how can you determine what it is?
  + Be familiar with some common file/directory functions, such as determining if the file exists.
  + Know how to use the with statement.
  + Be familiar with reading and writing CSV files, using the reader() and writer() methods.
* Be familiar with how the linear and binary searches work in general and their efficiency.
* Know how to handle exceptions, using the try-except-else-finally block, including standard as well as custom exceptions.
  + Be familiar with how to raise an exception and exit the program.
* Be able to generate random numbers using the randint() function.
* Know how to define and call a function, including passing arguments to the function and returning values from the function.
  + Know which objects are passed by value and which are passed by reference.
    - How can you make a mutable object immutable when passed to a function?
  + Know how to assign default arguments as well as an arbitrary number of keyword and non-keyword arguments in a function.
  + Can you return multiple values (i.e., unpacking)?
* Know how to pass and use command-line arguments with argv.
* Be familiar with how the selection and insertion sorts work in general.
  + Be familiar with the swap() function and how it works.
* Know how to declare and use a class, including instantiating a class object.
  + Be able to declare and use class and instance attributes and methods, as well as initialize attributes with the \_\_init\_\_() constructor.
  + Be able to add class customization by overriding the \_\_str\_\_() method.