The Midterm Will Cover Content From:

Material covered in Lectures #1-7 (up to and including ‘Logical Operators and Boolean Values’)
Chapters 1, 2, and 3 (except 3.7) of Python for Everybody
Practice Problems PP1, PP2, and PP3

Important topics to ensure you understand include: problem solving and computational thinking, variables (what they are, high level view of how they work, types), input and output, arithmetic operations and mathematical calculations, branching control structures (if, elif, else), Boolean expressions and logic operators.

Any questions regarding the midterm content, structure, practice questions, etc., should be posted in the Midterm and Exam Discussion board on cuLearn or on the course Discord server.

There will be less questions on the real midterm, as it will only be 1 hour long. These questions should give you an idea of what types of questions you could see.

Short answer

Write your answers for this question in the shortanswer.txt file.

1. What is a variable?
2. Give 3 examples of types of data that we have seen in class so far.
3. What does the assignment statement do? Give a small example.
4. What is a boolean expression? Give an example.
5. Python has several built-in type conversion functions. List two and show an example of each.
6. Why is indentation important in Python? What purpose does it serve?
7. List two comparison operators and give examples of how they are used.
8. Name a logical operator.

Long Answer

Write your answers for this question in the longanswer.txt file. **Note that you must show your work for each of these questions to receive any marks. An answer without demonstrating the steps involved will receive a grade of 0.**

1. Convert the decimal number 357 to binary.
2. Convert the binary number 0b1011011 to decimal
3. Simplify the logical expression below:

   not (((1 >= 1) and (3 == 4)) or ((not True) or (False)))
4. Simplify the logical expression below:

   not ((not (True and (not (False or True)))) or (5 < 5))

Writing Code

For these questions, you must write valid Python code to solve each problem. You should test your code to ensure it works correctly. Code that does not run will receive a grade of 0.

1. Write a Python program in the code-p1.py file that asks the user for a number (an integer) and then asks for three (3) words one at a time. After receiving all three words, the program prints each of the input words if their length is a multiple of the first input number. Note: Python contains a len(x) command, which evaluates to the number of characters in the string x.

   For example, if the user enters 3, ‘kitten’, ‘rewards’ and ‘cat’, the program would output ‘kitten’ and ‘cat’, because their lengths are each multiples of 3. The word ‘rewards’, with length 7, would not be printed.

2. Write a Python program in the code-p2.py file that ask the user to enter a student’s assignment, midterm, and exam grades. These grades may be decimal numbers. The program must compute and output the student’s final grade as a percentage using the weighting: assignment=30%, midterm=30%, exam=40%. The program must also print out whether the student passes or fails the course. For a student to pass the course, they must have both an exam grade of 50% or higher and an overall final grade of 60% or higher. For full marks, you must use a single if/else statement to print out whether the student passes or fails (i.e., no nested structures, no elif).

3. Write a Python program in the code-p3.py file that defines the following variables:

   ```
   guess = 73
   close_range = 10
   far_range = 20
   ```

   The program must ask the user to enter a guess. If the user enters the correct value (i.e., the value of the *guess* variable), the program must print out “Correct!”. If the user enters a value that is incorrect, but is less than *close_range* away from the correct value, the program must print out “Close but not quite correct.”. If the user enters a guess that is not correct/close but is within *far_range* from the correct value, the program must print out “Nice try, but you are far away”. If the users guess is not within *far_range* of the correct number, the program must print out “You are not very good at this.”.

4. Write a Python program in the code-p4.py file that asks the user to enter the X coordinate, Y coordinate, and name of three different locations (i.e., 9 different

values, 3 for each city). The program must then print out the names of the pair of cities that are closest to each other and the pair of cities that are furthest from each other. Remember, the distance between the point (x1, y1) and the point (x2, y2) can be calculated as:

$$dist = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$