ISE 5123 **Software Tools-Dec Support**
Spring 2020

Midterm Exam
March 31, 2020

NAME _____

Instructions :

1. Submit the codes for all problems in a notebook file on canvas.
2. All the solution steps will be evaluated while grading.
3. You have 75 minutes.
4. You may only use books, notes, slides, but all work is to be done on your own. The following statement must be typed in your notebook file in order for your test to be graded:

   "I understand that it is considered cheating to give or receive any unauthorized assistance on this online test."

1. **(20 points) True/False Section:**

   (a) Consider the following code segment:
   $count = 5$
   $whilecount > 1:$
   $print(count)$
   $Count- = 1$
   The output produced by this code is 5432.

   (b) Assume that the variable **data** refers to the list **[10, 20, 30]**. The expression **data[1:3]** evaluates to **[10, 20, 30]**.

   (c) The expression **list(map(math.sqrt, [9, 25, 36]))** evaluates to **[81, 625, 1296]**.

   (d) The recursive Fibonacci function makes approximately n^2 recursive calls for problems of a large size n.

   (e) Timing an algorithm with different problem sizes can give you an idea of the algorithm's run-time behavior on a particular hardware platform and a particular software platform.

2. **(30 points)** A local biologist needs a program to predict population growth. The inputs would be the initial number of organisms, the rate of growth (a real number greater than 0), the number of hours it takes to achieve this rate, and a number of hours during which the population grows. For example, one might start with a population of 500 organisms, a growth rate of 2, and a growth period to achieve this rate of 6 hours. Assuming that none of the organisms die, this would imply that this population would double in size every six hours. Thus, after allowing 6 hours for growth, we would have 1000 organisms, and after 12 hours, we would have 2000 organisms. Write a program that takes these inputs and displays a prediction of the total population.

3. **(30 points)** An alternative strategy for the **expo** function uses the following recursive definition:
   expo (number, exponent)
   =1, when exponent = 0
   = number * expo(number, exponent - 1), when exponent is odd
   = (expo $(number, exponent//2))^2$**, when exponent is even**

   Does this strategy reduce the computational complexity? Explain. You may use Big-O notation concept.

4. **(20 points)** Consider the assignment problem having the following cost table. The optimal solution is A-3, B-1, C-2. What input parameters are required to solve the problem using Python and Gurobi? Create the appropriate data structures in Python.

Person	**Job1**	**Job2**	**Job3**
A	5	7	4
B	3	6	5
C	2	3	4