Exam 3 Instructions

COP3035 – Intro to Python Programming

Instruction page – please read very carefully.

Date: Friday, July 12, 2024
Time Window: 9:00 AM - 9:00 PM

Format:
Location: This test is administered remotely; there's no need to come to the classroom.
The test will be accessible on Canvas during the designated window.
This is an open-book test. You may use textbooks, lecture notes, personal notes, formulae pages, handouts, other supplementary materials prepared in advance. These materials can be either paper or electronic format.
You might be required to write code and produce output. Ensure you have a working Python environment ready.
Individual Work: This test is meant to be completed independently. Collaboration is strictly prohibited. Do not discuss or share any details about the test or its solutions with anyone.

Submission:
Download the test and print it to answer. If you cannot print it, write your answers clearly on separate sheets of paper. (You can submit handwritten, typed, or mixed).
Clearly show and explain your work for each question, where necessary.
After completion, scan your test and submit it online via Canvas. Set aside at least 10 minutes for this process.
Use a scanning app to convert your test into a single PDF.
Ensure your submission is in the form of a single PDF file.
Clearly write your name and Z number on your test.
File Naming Convention: [Your Name] _[Z Number].pdf
While Canvas does allow multiple submissions, only the last one will be considered for grading.
Do Not share any information about the test or its solutions with others.
Exclude the instruction page from your submission.
Do not include your formula sheets in your test submission.

Note: Inquiries about test results, homework and extra credit grades must be sent to the instructors within 3 days after grades are published.

Please review these instructions thoroughly to ensure a smooth testing experience. Best of luck!

Exam 2

COP3035 – Intro to Python Programming

Name: ________________________________

zNumber: ______________________________

Question 1 (20 Points)

a) Write code that creates a list of the squares of integers from 0 to 9 using a list comprehension.

b) Write code that creates a list of the cubes of even integers from 0 to 20 using a list comprehension.

c) Given the following code that converts a number from binary to decimal, create the same output using a list comprehension.

```
    # Converting binary to decimal
    num = '111101011011'
    values = []
    for i,d in enumerate(num[::-1]):
        values.append(int(d)*2**i)
        #print(f' Digit:{d}, Index:{i}, Decimal:{value}')
    decimal = sum(values)
    print(f'{num} in binary is equal to {decimal} in decimal')
```

Question 2 (20 Points)

Using a dictionary comprehension, count the occurrences of each vowel in the given text. Consider both uppercase and lowercase vowels.

```
text = 'Life is like riding a bicycle. To keep your balance, you must keep moving.'
```

TIP: use .lower() and .count() built-in string functions.

Question 3 (20 Points)

Convert the following conditional structure to a ternary operation. Print and test both versions.

```
if num > 0:
    result = "Positive"
elif num == 0:
    result = "Zero"
else:
    result = "Negative"
```

Question 4 (20 Points)

Given the following function that returns the list of factors for an integer:

```
def factors(n):
    factors = []
    for i in range(1,n+1):
        #print(n,i,('Factor' if n%i == 0 else 'Not a factor'))
        factors.append(i) if n%i == 0 else None
    return factors
```

a) Write a function **isPrime(n)** that returns True or False if the number n is prime, using the **factors(n)** function. Remember, a number is prime if it has only 2 factors (1 and itself).

b) Write a function **primes(n)** that returns a list of all the primes up to the number n. Use the **isPrime(n)** function you defined.

c) Use the functions to return a report like the following (use another number):

```
# Example
The number 52 is prime: False
Factors: [1, 2, 4, 13, 26, 52]
Primes up to 52: [1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
Total number of primes up to 52: 16
```

Question 5: (20 Points)

Convert the following functions into lambda expressions. Test both the original and lambda versions and show your results.

Original functions	Lambda expressions

`def square(num):`
`    return num ** 2`	

`def concatenate(str1, str2):`
`    return str1 + str2`	

`def is_even(num):`
`    return num % 2 == 0`	

`def reverse(a):`
`    return a[::-1]`	

Question 6: (Bonus 10 Points)

Write a function to calculate the area of a rectangle given its width and height. Demonstrate how to call the function and display the output.