



JK Lakshmipat University, Jaipur
Institute of Engineering and Technology
Programming 1 (Python) (CS1139)
End Term, B.Tech., I Semester

Date: December 11, 2023

Duration: 3 hours

MM: 30

Course Instructor(s): Dr. Arpan Gupta, Dr. Surbhi Chhabra

There are 6 questions in this paper with sub-parts. You have to attempt all questions. Make appropriate assumptions wherever required. Your Python code should have proper comments, indentation, variable names and handle all the valid and invalid test cases.

1. Write the output of the following python code snippets? Assume that there are no indentation or spacing errors in the code. (1x8 - COs 2, 3, 4, 5, 6)

(a) `l = list(range(1, 11))
print(l[::-2])`

(b) `t = (1,2,4,3,8,9)
for s in range(1, len(t), 2):
 print(t[s], end="\t")`

(c) `a = 2.3
b = 1.3
c = a+b
if c == 3.6:
 print("Hello")
else:
 print("World")`

(d) `n = [1,2,3,4,5]
n1 = [x**2 for x in n if x%2==0]
mylst = [1,2,3,4,5]
new_lst = [x*x for x in mylst if x%2!=0]
print(new_lst + n1)`

(e) `a = "Hello World"
vowels = ["a","e","i","o","u"]
i = 0
while i < len(a):
 if a[i] in vowels:
 continue
 print(a[i])
 i+=1`

(f) `try:
 N = ["1.5", "0"]
 n1,n2 = int(N[0]),int(N[1])
 result = n1 / n2
 print(result)
except ValueError:
 print("Invalid input.")
except ZeroDivisionError:
 print("Cannot divide by zero.")
finally:
 print("Hello World!")`

(g) `a = "Data Scientist"
vowels=["a", "e", "i", "o", "u"]
i = 0
while i<len(a):
 if a[i] not in vowels:
 i+=1
 continue
 elif vowels.index(a[i])>1:
 i+=1
 continue
 print(a[i], end=" ")
 i=i+1`

```

(h) class Car:
    def _init_(self, make, model, year):
        self.make = make
        self.model = model
        self.year = year
        self.mileage = 0

    def display_info(self):
        print(f'{self.year}{self.make}{self.model}, \\
              Mileage: {self.mileage} miles')

    def drive(self, miles):
        self.mileage+=miles

# Creating instances of the Car class
car1 = Car("Toyota", "Camry", 2020)
car1.drive(50)
car1.display_info()
car2 = Car("Ford", "Mustang", 2019)
car2.drive(30)
car2.display_info()

```

2. (a) What does the following code compute for any two positive integer values?

(2 x 3 - COs 1, 2, 3, 4)

```

m = int(input("Enter a positive int "))
n = int(input("Enter a positive int "))
x, y = m, n
while True:
    if y>x:
        if x==0:
            break
        y, x = x, y-x
    else:
        if y==0:
            break
        x, y = y, x-y
    if x==0 or y==0:
        break
print("Result is : ", x)

```

- (b) What will be the output of the following code? Show the steps for all the recursive calls?

```

def rec(N):
    if N==0:
        return 0
    return 2*rec(N-1)+1

# calling rec function
print(rec(5))

```

(c) How many times does the print statement execute (as a function of N)?

```
for i in range(1, N+1):  
    for j in range(1, i+1):  
        print()        # how many times this statement executes
```

3. (a) What do you mean by inheritance in Object Oriented Programming? Give an example using Python? What are some of the types of inheritance?
- (b) What is the use of *enumerate()* function? Show with an example.
- (c) Mention the different modes for opening a file. Show an example for any one type of mode?

(2 x 3 - COs 3, 4, 5, 6)

4. Write an algorithm (pseudocode) and draw the flowchart for finding whether an input string is a palindrome or not. Consider the input string as an array of characters that is terminated by a NULL character ('\0'). A palindrom string is one, which is the same when read backwards. Eg. "ABCBA", or "MALAYALAM".

(3 - COs 1, 2)

5. Given two strings, *s* and *t*, write a Python program to determine if they are isomorphic. Note: Two strings *s* and *t* are isomorphic if the characters in *s* can be replaced to get *t*. All occurrences of a character must be replaced with another character while preserving the order of characters. No two characters may map to the same character, but a character may map to itself.

Example 1: Input: *s* = "egg", *t* = "add", Output: True

Example 2: Input: *s* = "foo", *t* = "bar", Output: False

Example 3: Input: *s* = "paper", *t* = "title", Output: True

(3 COs 4,5)

6. The N-Queen is the problem of placing N chess queens on an N×N chessboard so that no two queens attack each other. Note that a queen moves horizontally, vertically and diagonally on a chessboard. For example, Fig. 1 shows a valid solution of a 4 x 4 board, Figs. 2 and 3 show a valid and an invalid solution for 8 x 8 board.

You need to write a python program that will take a configuration of the queens as input and output "Valid" if the placement of a queens is valid else output "Invalid".

Input Format:

N - Input size of the board

Configuration for N=4

0 1 0 0

0 0 0 1

1 0 0 0

0 0 1 0

Output Format: "Valid" or "Invalid"

Constraints: N will be positive integer (≤ 10). The input configuration is in the form of 0s and 1s, where 0s denote an empty square on a chessboard, and 1s denote where a queen is placed. Assume there are no errors in the input provided.

(4 COs 1,2,3)

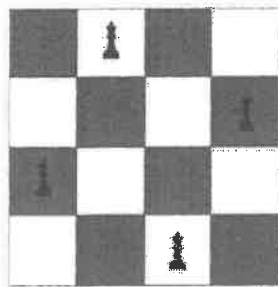


Figure 1: A valid 4 x 4 solution

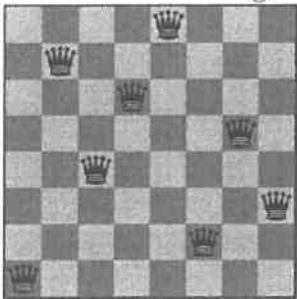


Figure 2: A valid solution for N=8

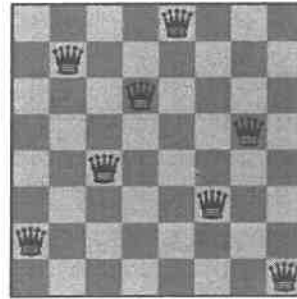


Figure 3: An invalid solution for N=8