Department of Electrical Engineering
Mid – Term Assignment Spring 2020
Date: 25/06/2020

Course Details

Course Title: Programming Fundamentals   **Module:** 02

Instructor: **Engr. Waqas khan**   **Total Marks:** 50

Student Details

Name: **FAWAD AHMAD**   **Student ID:**   **13204**

Q1.	(a)	Write Python code that asks the user for two integer values, it must then add the largest number times the smallest. The output should be in the following manner. Enter Number 1 : 5 Enter Number 2 : 3 Answer = 5+5+5 = 120	Marks 5 CLO 1
	(b)	You are asked to make a times table program in Python where a user will enter starting value, ending value and the value to find the times table for. For example Input Starting Value : 2 Input Ending Value : 4 Input Times Table : 4 4 x 2 = 8 4 x 3 = 12 4 x 4 = 16	Marks 5 CLO 1

Q2.	(a)	Write programs in Python to make the following shapes using LOOPS,
		
		a) ******
		          ******
		          ******
		
		b) *
		        **
		        ***
		        ****
		        *****
			Marks 14
			CLO 1

Q3.	(a)	Write a program in Python where a user is asked for 10 numbers, each number must be shown as ODD or EVEN respectively.	Marks 3 CLO 1
	(b)	You have the following python code, draw the flow chart of the whole code	Marks 3 CLO 1

```python
nterms = int(input("How many terms? "))
n1, n2 = 0, 1
count = 0
if nterms <= 0:
   print("Please enter a positive integer")
elif nterms == 1:
   print("Fibonacci sequence upto",nterms,":")
   print(n1)
else:
   print("Fibonacci sequence:")
   while count < nterms:
      print(n1)
      nth = n1 + n2
      n1 = n2
      n2 = nth
      count += 1
```

Question 1 (A):

Answer:

```
Sol :-

        Loop number = 0
        While loopnumber <= 4:
                                 2
            Print ('Please insert a number:, end = '')

        num = input()
        loop number != 1:
            if int (num) < int (smallest)
                    Smallest = num
            else:
                    Smallest = Smallest

        if loopnumber != 1:
                if int(num) > int (largest)
                    largest = num

        else:    largest = largest
        else
                Smallest = num
                largest = num
Print ('Done looping, smallest number is +str(smallest)
    +", largest number is " + " + "str (largest)).
```

Answer:

Sol : 1 (B)

→   Start = int (input ('Enter a Starting number : '))

→   end = int (input ('Enter an end number : '))

→   table = int (input ('Enter a table'))

→   While start < = end :

→   Print (table, 'x', start, '=' table * start)

→   Start + = 1

Answer:

Sol: 2 (a)

→   Print ("Print equilateral Triangle Pyramid using Stars")

→   Size = 5

→   m = (2* Size) - 2

→   for i in range (0, Size):

→   for j in range (0, m):

→   Print (end = " ")

→   m = m - 1 # decrementing m after each Loop

→   for j in range (0, i + 1):

→   # Printing full Triangle Pyramid using Stars

→   Print ("*", end = ' ')

→   Print (" ")

Answer:

Sol 2(B)

→   Steric = '* * * * * *'

→   for i in range (0,3):

→   Print (steric)

—— ✗ ——— ✗ ——— ✗ ——

Question 3 (A):

Answer:

Q 3 (A)

```
→    Number = [ ]

→    n = int (input ("Enter number of elements:\t"))

→    for i in range (1, 10):

→        all Element = int (input ("Enter element : "))

→        numbers.append (all Element)

→    even lst = [ ]

→    odd lst = [ ]

→    for j in numbers:

→        if j % 2 == 0:

→            even lst.append (j)

→        else:

→            odd lst.append (j)

→    Print ("Even number list\t", even_lst)

→    Print ("odd number list\t", odd_lst)
```

Answer:

```
                    ┌──────────────┐
                    │    Start     │
                    └──────┬───────┘
                           ▼
                    ┌──────────────┐
                    │   n₁ = 0     │
                    └──────┬───────┘
                           ▼
                    ┌──────────────┐
                    │   n₂ = 1     │
                    └──────┬───────┘
                           ▼
                    ┌──────────────┐
                    │  Count = 0   │
                    └──────┬───────┘
                           ▼
                    ┌──────────────┐
                    │   fib = 0    │
                    └──────┬───────┘
                           ▼
```

$n_1 = 0$

$n_2 = 1$

$Count = 0$

$fib = 0$

nterms <= 0 → NO → END

yes

output Fibonacci $(n_1) = fib$

output carriage return and linefeed

$nth = n_1 + n_2$

$n_1 = n_2$

$n_2 = nth$

$Count += 1$