Inspiring Excellence

Course Title: Programming Language I

Course Code: CSE 110

Assignment no: 4

Class Tasks	5
Home Tasks	5 (50 Points)
Total Tasks	10

Class Task 1

CW-1.1) Write a function called **even_checker** that takes a number as its argument and prints whether the number is even or odd **inside the function**.

Sample Function Call	Output
`even_checker(10)`	`Even!!`
`even_checker(17)`	`Odd!!`

CW-1.2) Write a function called **is_even()** that takes a number as an argument and **returns** True if the number is even otherwise **returns** False.

Sample Function Call	Output
`result = is_even(10)` `print( result )`	`True`
`result = is_even(17)` `print( result )`	`False`

CW-1.3) Write a function called **is_pos()** that takes a number as an argument and **returns** True if the number is positive otherwise **returns** False.

Sample Function Call	Output
`result = is_pos(-5)` `print( result )`	`False`
`result = is_pos(12)` `print( result )`	`True`

CW-1.4) Write a function called **sequence()** that takes an integer in its parameter called n. Now, if n is **positive** then it prints all the **even** numbers from **0 to n**, otherwise if n is **negative** it prints all the **odd** numbers from **n to -1**.
Note: **You must call** the functions from **CW-1.2** and **CW-1.3**, otherwise this task would be considered invalid.

Sample Function Call	Output	Explanation
`sequence(10)`	`0 2 4 6 8 10`	Here, 10 is positive so 0,2,4,6,8,10 were printed.
`sequence(-7)`	`-7 -5 -3 -1`	Here, -7 is negative so -7,-5,-3,-1 were printed.
`sequence(7)`	`0 2 4 6`	Here, 7 is positive so 0,2,4,6 were printed
`sequence(-8)`	`-7 -5 -3 -1`	Here, -8 is negative so -7,-5,-3,-1 were printed.

Class Task 2

CW-2.1) Write a function called **is_valid_triangle** that takes 3 integer numbers as arguments. The function will **return** True if the 3 sides can form a valid triangle otherwise it'll **return** False.

Note: In a valid triangle, the sum of any two sides will be greater than the third side.

Sample Function Call	Output	Explanation
`result=is_valid_triangle(7,5,10)` `print( result )`	`True`	Here, 7+5>10, 5+10>7 also, 10+7>5. Thus, these 3 sides can form a valid triangle.
`result=is_valid_triangle(3,2,1)` `print( result )`	`False`	Here, 1+2<=3, thus, these 3 sides can NOT form a valid triangle.

CW-2.2) Write a function called **tri_area** that takes 3 sides of a triangle as 3 arguments. The function should calculate and print the area of the triangle only if it's a valid triangle otherwise print that it's not a valid triangle.
Area of triangle = √[s(s−a)(s−b)(s−c)], where 's' is the semi perimeter of the triangle. So, semi-perimeter = perimeter/2 = (a + b + c)/2.
Note: **You must call** the function written in task **CW-2.1**, otherwise this task would be considered invalid.

Sample Function Call	Output	Explanation
`tri_area(3,2,1)`	`Can't form triangle`	Here, 1+2<=3, thus, these 3 sides can NOT form a valid triangle.
`tri_area(7,5,10)`	`16.248`	Here, 7,5,10 is able to form a valid triangle so, using the formula we get the area as 16.248

Class Task 3

CW-3.1) Write a function called **circle** that takes a radius in its parameter and **returns** the **area** of the circle. [**Note:** area of a circle is πr^2**]**

Sample Function Call	Output
`circle(5)`	`78.5398`

CW-3.2) Write a function called **sphere** that takes a radius in its parameter and **returns** the **volume** of the sphere. **[Note:** area of a sphere is $\frac{4}{3}\pi r^3$**]**

Sample Function Call	Output
`sphere(5)`	`523.5988`

CW-3.3) Write a function called **fitting** that takes three integer values as arguments. First two arguments are the **two diameters** and the third argument is the **dimension**. If it is 3 dimensional then you must calculate the volume of the spheres otherwise calculate the area of the circles. Finally, find out and print which circle/sphere can fit inside which one? Also, print how much space would be left if the smaller circle/sphere is put inside the larger one?

Note: **You must call** the function written in task **CW-3.1 & CW-3.2**, otherwise this task would be considered invalid. Also, you can assume only 2 or 3 dimensions can be passed.

Sample Function Call	Output
`fitting(8,10,2)`	`Circle-1 can fit inside Circle-2 and 28.274 square units would be left.`
`Explanation:`	`Since the dimension is 2 we need to calculate the area. Here, the area of Circle-1 is less than Circle-2 so Circle-1 can fit inside Circle-2. Moreover, we can find out how much square units will be left by subtracting the area.`
`fitting(30,14,3)`	`Sphere-2 can fit inside Sphere-1 and 12700.412 cubic units would be left.`
`Explanation:`	`Since the dimension is 3 we need to calculate the volume. Here, the volume of Sphere-2 is less than Sphere-1 so Sphere-2 can fit inside Sphere-1. Moreover, we can find out how much cubic units will be left by subtracting the volume.`
`fitting(5,5,3)`	`Impossible to fit.`
`Explanation:`	`Since the diameters are the same, it will be impossible to fit one shape inside another.`

Class Task 4.1

Trace the following code to generate outputs

1	`print("Starting.....3")`	OUTPUT
2	`c=99`	
3	`def F3(a):`	
4	`    print("F3 begun")`	
5	`    print(a+c)`	
6	`    print("F3 ending")`	
7	`    return a*5`	
8	`print("Starting...2")`	
9	`def F1():`	
10	`    print("F1 begun")`	
11	`    F2(5)`	
12	`    print("F1 ended")`	
13	`print("Starting..1")`	
14	`def F2(c):`	
15	`    print("F2 begun")`	
16	`    print( F3(c)+30 )`	
17	`    print("F2 ended")`	
18	`print(F1())`	
19	`print("All Done")`	

Class Task 4.2

[FACTORIAL RECURSION TRACING]

Trace the following code to generate outputs

1	`def factorial(n):`	OUTPUT
2	`    if n==0 or n==1:`	
3	`        return 1`	
4	`    else:`	
5	`        return n*factorial(n-1)`	
6	`print(factorial(5))`	

Class Task 5

CW-5.1) Write a function called one_to_N that prints 1 till N recursively.
Hint: N is a number taken as input from the user and you need to print the numbers starting from 1 to N recursively.

Sample Input	Sample Function Call	Output
n=5	one_to_N(1,n)	1 2 3 4 5
n=11	one_to_N(1,n)	1 2 3 4 5 6 7 8 9 10 11

CW-5.2) Write a function reverse_printing from N to 1 recursively

Hint: N is a number taken as input from the user and you need to print the numbers starting from N to 1.

Sample Input	Sample Function Call	Output
n=6	reverse_printing(1,n)	6 5 4 3 2 1
n=3	reverse_printing(1,n)	3 2 1

CW-5.3) Write a function summation to sum till N recursively

Hint: N is a number taken as input from the user and you need to add the numbers starting from 1 to N recursively and print the sum.

Sample Input	Sample Function Call	Output
`n=4`	`recursive_summation(1,n)`	`10`
`n=12`	`recursive_summation(1,n)`	`78`

Home Task 1

HW-1.1) Write a function called **is_prime** which takes an integer in its parameter to check whether a number is prime or not. If the number is prime then the function returns **True** otherwise it returns **False**. **[3 points]**

Sample Function Call	Output
`prime_check = is_prime(7)` `print(prime_check)`	`True`
`prime_check = is_prime(15)` `print(prime_check)`	`False`

HW-1.2) Write a function called **is_perfect** which takes an integer in its parameter to check whether a number is perfect or not. If the number is perfect then the function returns **True** otherwise it returns **False**. **[3 points]**

Sample Function Call	Output
`perfect_check = is_perfect(6)` `print(perfect_check)`	`True`
`perfect_check = is_perfect(33)` `print(perfect_check)`	`False`

HW-1.3) Write a function called **special_sum** that calculates the sum of all numbers that are either prime numbers or perfect up till the integer value given in its parameter. This integer value must be taken as user input and passed into the function. **[4 points]**

Note: **You must call** the functions written in task **HW-1.1 & HW-1.2**, otherwise this task would be considered invalid.

Sample Input	Sample Function Call	Output
`8`	`result = special_sum(8)` `print(result)`	`23`
`Explanation:`	`Between 1 to 8 the Prime numbers are 2,3,5,7 and 6 is a Perfect number. So, the summation is 2+3+5+7+6=23.`	

30	`result = special_sum(30)` `print(result)`	163
Explanation:	Between 1 to 30 the Prime numbers are 2,3,5,7,11,13,17,19,23,27,29 and the Perfect numbers are 6,28. So, the summation is 2+3+5+7+11+13+17+19+23+29+6+28=163.	

Home Task 2

HW-2.1) Write a simple function called **show_dots** that takes a number as an argument and then prints that amount of dots inside the function. **[2 points]**

Note: You can use `print(,end="")` to avoid the next output being printed on the next line.

Sample Function Call	Output
`show_dots(5)`	.....
`show_dots(3)`	...

HW-2.2) Write a function called **show_palindrome** that takes a number as an argument and then prints a palindrome inside the function. **[4 points]**

Note: You can use `print(,end="")` to avoid the next output being printed on the next line

Sample Function Call	Output
`show_palindrome(5)`	123454321
`show_palindrome(3)`	12321

HW-2.3) Write a function called **show_triangle** that takes a number as an argument and then prints a **palindromic triangle**. Moreover, the empty spaces surrounding the triangle are filled with dots(**.**) .

Note: **You must call** the functions written in task **HW-2.1 & HW-2.2**, otherwise this task would be considered invalid. **[4 points]**

Sample Function Call	Output

`show_triangle(5)`	`....1....`
	`...121...`
	`..12321..`
	`.1234321.`
	`123454321`

`show_triangle(3)`	`..1..`
	`.121.`
	`12321`

Home Task 3

HW-3.1) Write a function called **calc_tax** that takes 2 arguments which are **your age** then **your salary**. The function must calculate and **return** the tax as the following conditions:

- No tax if you are less than 18 years old.
- No tax if you get paid less than 10,000
- 7% tax if you get paid between 10K and 20K
- 14% tax if you get paid more than 20K **[4 points]**

Sample Function Call	Output	Explanation
`t = calc_tax(16,20000)` `print(t)`	`0`	Here, the age is less than 18 so 0 tax.
`t = calc_tax(20,18000)` `print(t)`	`1260`	Here, the age is greater than 18 and income is between 10K-20K so tax is 7% of 18000 = 1260.

HW-3.2) Write a function called **calc_yearly_tax** that takes no arguments. Inside the function it should take **first input as your age** and then **12 others inputs** as income of each month of the year. The function must calculate and print Tax for each month and finally print the total Tax of the whole year based on the **HW-3.1 conditions**. **[6 points]**

Note: You must call the function written in task **HW-3.1**, otherwise this task would be considered invalid.

Sample Function Call	Sample Input	Output
`calc_yearly_tax()`	22 8000 15000 22000 2300 15300 21000 34000 9000 27000 88000 32000 7300	Month1 tax: 0 Month2 tax: 1050.0 Month3 tax: 3080.0 Month4 tax: 0 Month5 tax: 1071.0 Month6 tax: 2940.0 Month7 tax: 4760.0 Month8 tax: 0 Month9 tax: 3780.0 Month10 tax: 12320.0 Month11 tax: 4480.0 Month12 tax: 0 Total Yearly Tax: 33481.0

Home Task 4

HW-4.1) Write a recursive function called **reverse_digits** that takes an integer n as an argument and prints the digits of n in reverse order.

Hint: Think about how you solved it using loop **[5 points]**

Sample Input	Sample Function Call	Output
12345	`reverse_digits(n)`	5 4 3 2 1
649	`reverse_digits(n)`	9 4 6
1000	`reverse_digits(n)`	0 0 0 1

HW-4.2) Write a recursive function called **power** that takes two arguments a and n. The function calculates and returns the value of a^n. **[5 points]**

Sample Input	Sample Function Call	Output
3 5	power(a, n)	243
9 4	power(a, n)	6561

Home Task 5

HW-5.0) Write a function called **sequence_iterative** that uses loop to calculate the value of **y** if the expression of y is as follows (Here, N is the input)

`y = 1 - 2 + 3 - 4 + 5 - ..... + N` **[2 points]**

Sample Input	Sample Function Call	Output
5	sequence_iterative(5)	3
10	sequence_iterative(10)	-5

HW-5.1) Write a **recursive** function called **sequence_recursive** that calculates the value of **y** if the expression of y is as follows (Here, N is the input)

`y = 1 - 2 + 3 - 4 + 5 - ..... + N` **[5 points]**

Sample Input	Sample Function Call	Output
5	sequence_recursive(5)	3
10	sequence_recursive(10)	-5

HW-5.2) Trace the Recursion code of HW-5.1 assuming the input value is 6. **[3 points]**