

ASSIGNMENT 1

1) Even or Odd: Write a function called evenOrOdd that takes an integer as input and returns "Even" if the number is even, and "Odd" if the number is odd.

Python Code

```
def even_or_odd(number):  
    if number % 2 == 0:  
        return "Even"  
    else:  
        return "Odd"
```

Getting user input

```
number = int(input("Enter an integer:"))
```

Calling the function.

```
result = even_or_odd(number)  
print(f"The number {number} is {result}.")
```

2)

Maximum of Two Numbers: Write a function called maxOfTwo that takes two numbers as input and returns the maximum of two.

Python Code

```
def maxOfTwo(num1, num2):  
    if num1 > num2:  
        return num1  
    else:  
        return num2
```

Getting user input

```
number1 = float(input("Enter the first number:"))
```

```
number2 = float(input("Enter the second number:"))
```

You may delay, but time will not.

11 Calling out the max OF Two Function
 $\text{max_number} = \text{max OF Two}(\text{number1}, \text{number2})$

print the result.
print(f "The maximum of {number1} and {number2} is
{max_number}")

③ Leap Year Checker: Write a Function called isLeapYear that takes a year as input and returns True if it is a leap year, and False otherwise. (A leap year is divisible by 4, but not by 100 unless it is also divisible by 400).

Python Code:

def isLeapYear(year):

if (year % 4 == 0) and ((year % 100 != 0 or year % 400 == 0))
return True

else:

return False

Getting user input for the year.

year = int(input("Enter a year :"))

Call if isLeapYear Function

If isLeapYear(year):

print(f "Year {year} is a leap year.")

else:

print(f "Year {year} is not a leap year.")

Success has a simple formula do your best, and people may like it.

(A) Factorial Calculator: 1. Write a function called Factorial that takes an integer as input and returns its factorial.
The Factorial of a number n is the product of all positive integers less than or equal to n.

Python Code

```
def factorial(number):
```

```
    if number < 0:
```

```
        return "Factorial is not defined for negative number"
```

```
    elif n == 0:
```

```
        return 1
```

```
    else:
```

```
        product = 1
```

```
        for i in range(1, number + 1):
```

```
            product *= i
```

```
        return product
```

```
number = int(input("Enter a non-negative number:"))
```

```
# Call the Factorial function to calculate the result.
```

```
result = factorial(number)
```

```
# Printing the result
```

```
print(f"The Factorial of {number} is: {result}")
```



You may delay, but time will not.

⑤ Greatest Common Divisor (GCD). Write a function called gcd that takes two positive integers as input and returns their greatest common divisor.

Python Code

```
def gcd(a, b):
```

handling invalid input (non-positive integers)

if $a \leq 0$ or $b \leq 0$:

return "Invalid input please return a positive integer!"

while $b \neq 0$:

remainder = $a \% b$

$a = b$

$b = \text{remainder}$

return a .

Getting the user input of two positive integers.

number1 = int(input("Enter the first number:"))

number2 = int(input("Enter the second number:"))

Calling out the function:

```
greatest_divisor = gcd(number1, number2)
```

Print the Result

```
if isinstance(greatest_divisor, str):  
    print(greatest_divisor)
```

else:

```
print(f"The greatest common divisor of {number1} and  
{number2} is {greatest_divisor}")
```

Success has a simple formula: do your best, and people may like it.

⑥ Absolute Value: Write a Function called absolute that
takes a number as input and returns its absolute value using
the built-in abs() function.

+ def absoluteValue(num) code
 if num >= 0
 return num
 else :
 return -num.

user input
number = float(input("Enter a number : "))
absolute_value = absoluteValue(number)
print(f"The absolute value of a number is {absolute_value}")



You may delay, but time will not.

④ Temperature Conversion: Write a Function called ConvertTemperature that takes in Celcius and as user input returns it converted to Fahrenheit. The conversion formula to Fahrenheit = $(\text{Celsius} \times \frac{9}{5}) + 32$.

Python Code

```
def convertTemperature(Celsius):  
    fahrenheit = (Celsius * 9/5) + 32  
    return fahrenheit
```

User input

```
Celsius_temp = float(input("Enter the temperature in Celcius:"))  
fahrenheit_temp = convertTemperature(Celsius_temp)  
print(f"\n{Celsius_temp} degree Celcius is equal to {fahrenheit_temp} degrees Fahrenheit")
```

and

⑧ Grade Calculator: Write a function called CalculateGrade that takes a score as input and returns the corresponding letter grade according to the following schema.

Python Code:

90-100 : A

80-89 : B

70-79 : C

60-69 : D

Below 60 : F

Python Code:

def calculateGrade(score):

This function takes score as user input and returns the corresponding letter grade.

if score >= 90 :

 return "A"

elif score >= 80 :

 return "B"

elif score >= 70 :

 return "C"

elif score >= 60 :

 return "D"

else :

 return "F"

While True :

try :

 score = float(input("Enter your score (0-100)"))

if 0 <= score <= 100

 break



You may delay, but time will not.

else:

print("Invalid score: please enter a value between 0 and 100.
except ValueError:

print("Invalid input. Please enter a number.")

Calculating and printing the grade

grade = calculateGrade(score)

print("Your grade is:", grade)

9) Sum of square with loop. Write a function called sumOfSquares that takes a positive integer n as input and calculates the sum of the squares of all positive integers up to n using a loop. The function should return the sum.

Python Code

```
def sumOfSquares(n):
    # This function calculates the sum of squares of all positive
    # integers up to n.
    # Returns the sum of all positive integers up to n.

    # Looping through all the positive integers.
    for i in range(1, n+1):
        # Square the current number and add it to the sum.
        sum += i * i
    return sum
```

Get user input
while True:

try:

n = int(input("Enter a positive integer: "))

if n > 0:

break

else:

print("Invalid input. Please enter a positive integer.")

except ValueError:

print("Invalid input. Please enter a positive integer.")

You may delay, but time will not.

Calculate and print the sum of squares.

sum_of_squares = sum_of_squares(n)

print("The sum of squares of all positive integers up to
n is: ", sum_of_squares)

(10)

Quadratic Equation Solver: Write a function called solveQuadratic that takes three numbers $\{a, b, c\}$ as user input representing the coefficients of a quadratic equation $ax^2 + bx + c = 0$ and returns the solution as a tuple. If the equation has no real roots, return None.

Python Code:

def solveQuadratic(a, b, c):

The function solves a quadratic equation of the form
 $ax^2 + bx + c = 0$ and returns a tuple.

a: The coefficient of the x^2 term.

b: The coefficient of x term.

c: The constant term.

Returns a tuple containing Real part if it exists or
imaginary None if does not exist.

$$\text{discriminate} = b**2 - 4*a*c$$

Checking for real roots based on discriminant.

If discriminant > 0

Two real roots.

$$\text{root1} = (-b + \text{discriminate}**0.5)$$

$$\text{root2} = (-b - \text{discriminate}**0.5)$$

return root1, root2.

Success has a simple formula: do your best, and people may like it.

else:

No real roots (Complex roots)

return None

Getting user input:

while True:

try:

a = float(input("Enter the Coefficient of x^2 term (a):"))

b = float(input("Enter the Coefficient of x term (b):"))

c = float(input("Enter the Constant (c):"))

break

except ValueError:

print("Invalid Input. Please Enter numbers only.")

Solving quadratic and printing.

Solutions = solveQuadratic(a, b, c)

If Solutions is not None:

print("The solutions are", solutions[0], "and", solutions[1])

else:

print("The Solution has no real roots.")