Python Assignment

14-02-2025
Suneetha.V

1.Write a Python program to calculate the area of a rectangle using user input for length and width.

Ans:----------

```
 # Input: Length and width of the rectangle

    length = float(input("Enter the length of the rectangle: "))
    width = float(input("Enter the width of the rectangle: "))

# Calculate the area of the rectangle
    area = length * width

#  Display the area
     print(f"The area of the rectangle is: {area}")

#input: Enter the length of the rectangle: 5
          Enter the width of the rectangle: 3
```

final output: The area of the rectangle is: 15.0

2.Write a Python program to find the maximum of three numbers using conditional statements.

Ans:----------

```
# Input: Three numbers:
a = float(input("Enter the first number: "))
b = float(input("Enter the second number: "))
c = float(input("Enter the third number: "))

# Conditional statements to find the maximum:
if a >= b and a >= c:
    maximum = a
elif b >= a and b >= c:
    maximum = b
else:
    maximum = c

#  Display the maximum value:
print("The maximum number is:", maximum)
```

Python Assignment

#input: Enter the first number: 10
Enter the second number: 20
Enter the third number: 15

final output: The maximum number is: 20.0

3..Write a Python program to swap the values of two variables without using a temporary variable.

Ans:----------

```
# Input: Two variables
a = int(input("Enter the value of a: "))
b = int(input("Enter the value of b: "))

# Swapping values without a temporary variable
a, b = b, a

# Display swapped values
print(f"After swapping: a = {a}, b = {b}")
```

#input: Enter the value of a: 5
Enter the value of b: 10

final output: After swapping: a = 10, b = 5

4.Write a Python program to convert temperature from Celsius to Fahrenheit and vice versa using functions.

Ans:----------

```
# Function to convert Fahrenheit to Celsius
    def fahrenheit_to_celsius(fahrenheit):
      return (fahrenheit - 32) * 5/9

# Input: Ask the user for the temperature and conversion type
  temp_value = float(input("Enter the temperature value: "))
conversion_type = input("Enter the conversion type (C to F or F to C):
").strip().lower()
```

Python Assignment

```
# Perform the conversion based on user input
      if conversion_type == 'c to f':
            fahrenheit = celsius_to_fahrenheit(temp_value)
       print(f"{temp_value}°C is equal to {fahrenheit}°F")
    elif conversion_type == 'f to c':
          celsius = fahrenheit_to_celsius(temp_value)
       print(f"{temp_value}°F is equal to {celsius}°C")
    else:
      print("Invalid conversion type! Please enter 'C to F' or 'F to C'.")

#input:  Enter the temperature value: 25
           Enter the conversion type (C to F or F to C): c to f
```

final output: 25.0°C is equal to 77.0°F

5.Write a Python program to count the number of vowels in a given string.

Ans:----------

```
# Function to count vowels in a string
    def count_vowels(string):
         vowels = "aeiouAEIOU"  # List of vowels (both lowercase and uppercase)
         count = 0                        # Initialize count to 0

 # Loop through each character in the string
    for char in string:
        if char in vowels:
            count += 1                  # Increment count if the character is a vowel

     return count

# Input: String from the user
   input_string = input("Enter a string: ")

# Count vowels in the input string
    vowel_count = count_vowels(input_string)

# Print the result
print("The number of vowels in the given string is: {vowel_count}")
```

Python Assignment

#input : string is "Hello World"

final output : The number of vowels in the given string is: 3

6.Write a Python program to check if a given number is prime or not.

ans:----------

```
# Function to check if a number is prime
    def is_prime(n):
        if n <= 1:
        return False              # Numbers less than or equal to 1 are not prime
   for i in range(2, n):
      if n % i == 0:
        return False              # n is divisible by i, so it's not prime
   return True                    # n is prime if no divisors were found

# Input: Number to check
num = int(input("Enter a number: "))      # number is 5

# Check and print result
if is_prime(num):
   print("{num} is a prime number.")
else:
   print("{num} is not a prime number.")
```

final output: 5 is a prime number.

7.Write a Python program to find the factorial of a given number using recursion.

Ans:----------

```
# Function to find the factorial using recursion
def factorial(n):
   # Base case: If n is 0 or 1, return 1 (since 0! = 1 and 1! = 1)
   if n == 0 or n == 1:
      return 1
# Recursive case: n * factorial of (n-1)
   else:
      return n * factorial(n - 1)
```

Python Assignment

```
# Input: Number to find the factorial of
num = int(input("Enter a number: "))        # number is 5

# Calculate and print the factorial
print(f"The factorial of {num} is {factorial(num)}")
```

final output: The factorial of 5 is 120

8.Write a Python program to generate the Fibonacci sequence up to a certain number of terms.

Ans:----------

```
# Function to generate Fibonacci sequence
   def fibonacci(n):
 # Initialize the first two terms
    a, b = 0, 1
    for _ in range(n):
        print(a, end=" ")         # Print the current term
        a, b = b, a + b           # Update a and b for the next term

# Input: Number of terms
num_terms = int(input("Enter the number of terms: "))   #10

# Generate the Fibonacci sequence
  fibonacci(num_terms)
```

final output: 0 1 1 2 3 5 8 13 21 34

9.Write a Python program to remove duplicates from a list.

Ans:----------

```
# Given list with duplicates
   my_list = [1, 2, 3, 4, 5, 2, 3, 6, 7, 4]

# Remove duplicates by converting the list to a set and back to a list
   my_list = list(set(my_list))
```

Python Assignment

```
# Print the list without duplicates
   print("List without duplicates:", my_list)
```

final output List without duplicates: [1, 2, 3, 4, 5, 6, 7]

10.Write a Python program to find the intersection of two lists.

Ans:----------

```
# Given two lists
list1 = [1, 2, 3, 4, 5]
list2 = [4, 5, 6, 7, 8]

# Find the intersection using set
intersection = list(set(list1) & set(list2))

# Print the intersection
print("The intersection of the two lists is:", intersection)
```

final output The intersection of the two lists is: [4, 5]

11.Write a Python program to find the longest word in a given list of words.

Ans:----------

```
# Given list of words
words = ["apple", "banana", "grape", "kiwi", "strawberry"]

# Find the longest word using max() with key=len
longest_word = max(words, key=len)

# Print the longest word
print("The longest word is:", longest_word)
```

final output: The longest word is: “strawberry”

12.Write a Python program to count the occurrences of each word in a given string.

Python Assignment

Ans:----------

```
# Given string
input_string = "hello world hello hello"

# Split the string into words
words = input_string.split()

# Create a dictionary to store the count of each word
word_count = {}

# Count occurrences of each word
for word in words:
    word_count[word] = word_count.get(word, 0) + 1

# Print the word counts
print(word_count)
```

final output: {'hello': 3, 'world': 1}

13.Write a Python program to reverse a given string.

Ans:----------

```
# Given string
input_string = "Hello, World!"

# Reverse the string using slicing
reversed_string = input_string[::-1]

# Print the reversed string
print("Reversed string:", reversed_string)
```

final output: Reversed string is: !dlroW ,olleH

14.Write a Python program to sort a list of tuples based on the second element of each tuple.

Ans:----------

Python Assignment

```
# Create a list of tuples
my_list = [(1, 3), (2, 1), (3, 2), (4, 4)]

# Sort the list of tuples based on the second element of each tuple
sorted_list = sorted(my_list, key=lambda x: x[1])

# Print the sorted list
print("List sorted by the second element of each tuple:", sorted_list)
```

final output is: List sorted by the second element of each tuple: [(2, 1), (3, 2), (1, 3), (4, 4)]

15.Write a Python program to find the sum of all elements in a list using a loop.

Ans:----------

```
# Create a list of numbers
my_list = [1, 2, 3, 4, 5]

# Initialize a variable to store the sum
total_sum = 0

# Loop through the list and add each element to total_sum
for num in my_list:
    total_sum += num

# Print the result
print("The sum of all elements in the list is:", total_sum)
```

final output is: The sum of all elements in the list is: 15

16.Write a Python program to remove the last element from a list.

Ans:----------

```
# Create a list
my_list = [1, 2, 3, 4, 5]
```

Python Assignment

```
# Remove the last element from the list
my_list.pop()

# Print the modified list
print("List after removing the last element:", my_list)
```

final output is: List after removing the last element: [1, 2, 3, 4]

17.Write a Python program to check if a given string is a palindrome.

Ans:----------

```
def is_palindrome(s):
# Compare the string with its reverse
    return s == s[::-1]

# Example usage
input_string = "racecar"
if is_palindrome(input_string):
    print(" is a palindrome.")
else:
    print(" is not a palindrome.")
```

The final output is: 'racecar' is a palindrome.

18.Write a Python program to find the common characters between two strings.

Ans:----------

```
    def common_characters(str1, str2):
    # Convert both strings to sets to remove duplicates and find intersection
    set1 = set(str1)
    set2 = set(str2)

    # Find common characters using set intersection
    common_chars = set1.intersection(set2)

# Example usage
str1 = "apple"
```

Python Assignment

```
str2 = "orange"

result = common_characters(str1, str2)
```

final output: Common characters: ['a', 'e']

19.Write a Python program to find the length of the longest consecutive sequence of a given list of integers.

Ans:----------

```
def longest_consecutive(nums):
    if not nums:
        return 0

 nums = set(nums)                    # Remove duplicates by converting to a set
 longest_streak = 0

   for num in nums:
       if num - 1 not in nums:              # Check if it's the start of a sequence
          current_num = num
          current_streak = 1

          while current_num + 1 in nums:    # Check the consecutive sequence
              current_num += 1
              current_streak += 1

          longest_streak = max(longest_streak, current_streak)

   return longest_streak

# Example usage

nums = [100, 4, 200, 1, 3, 2]
result = longest_consecutive(nums)
print(f"The length of the longest consecutive sequence is: {result}")
```

final output: The length of the longest consecutive sequence is: 4

20.Write a Python program to find the difference between two sets.

Python Assignment

Ans:----------

```
# Create two sets
        set1 = {1, 2, 3, 4, 5}
        set2 = {4, 5, 6, 7, 8}

# Find the difference between set1 and set2
 difference = set1 - set2                          # Using the '-' operator

# difference = set1.difference(set2)

# Print the result
print("The difference between set1 and set2 is:", difference)

#final output:  {1,2,3}
```

21.Arithmetic Operators

- Create two variables a and b with numeric values.
- Calculate the sum, difference, product, and quotient of a and b.
- Print the results.
- Ans:----------

```
# Create two variables with numeric values
a = 20
b = 5

# Calculate the sum, difference, product, and quotient

sum_result = a + b                  # Sum of a and b
difference_result = a - b           # Difference of a and b
product_result = a * b              # Product of a and b
quotient_result = a / b             # Quotient of a and b (float)

#print the result:

print("the sum of two numbers  is:",  sum)                    output 25
print("the difference of two numbers is:", difference)       output 15
print("the product of two numbers is:", product)             output 100
print("the quotient of two numbers is:", quotient)           output 4.0
```

Python Assignment

22. Comparison Operators

- Compare the values of a and b using the following comparison operators: <, >, <=, >=, ==, and !=.
- Print the results of each comparison.
- Ans:----------
- # Initialize two variables
- a = 15
- b = 10
- # Using comparison operators
- less_than = a < b # Check if a is less than b
- greater_than = a > b # Check if a is greater than b
- less_than_equal = a <= b # Check if a is less than or equal to b
- greater_than_equal = a >= b # Check if a is greater than or equal to b
- equal = a == b #Check if a is equal to b
- not_equal = a != b # Check if a is not equal to b

print the result of each comparison

- print(a<b) output false
- print(a>b) output true
- print(a<=b) output false
- print(a>=b) output true
- print(a==b) output false
- print(a!=b) output true

23.Logical Operators

- Create two boolean variables, x and y.
- Use logical operators (and, or, not) to perform various logical operations on x and y.
- Print the results.

Ans:----------

- # Create two boolean variables x and y
- x = True
- y = False

Python Assignment

```
# Using logical operators to perform operations

and_result = x and y      # True if both x and y are True, otherwise False
or_result = x or y        # True if either x or y is True
not_x = not x             # True if x is False, and False if x is True
not_y = not y             # True if y is False, and False if y is True

# Print the results

print("x and y:", and_result)    # Result of x and y     output False
print("x or y:", or_result)      # Result of x or y      output True
print("not x:", not_x)           # Result of not x       output False
print("not y:", not_y)           # Result of not y       output True
```

24: Assignment Operators

- Create a variable total and initialize it to 10.
- Use assignment operators (+=, -=, *=, /=) to update the value of total.
- Print the final value of total.

- Ans:----------

```
# Create a variable 'total' and initialize it to 10
total = 10

# Use assignment operators to update the value of 'total'

total += 5          # Adds 5 to total (total = total + 5)              output : 15
total -= 3          # Subtracts 3 from total (total = total - 3)       output : 12
total *= 2          # Multiplies total by 2 (total = total * 2)        output  : 24
total /= 4          # Divides total by 4 (total = total / 4)           output : 6.0

# Print the final value of total
print("The final value of total is:", total)
```

#final output : 6.0

Python Assignment

25: Bitwise Operators (Optional)

- If you are comfortable with bitwise operators, perform some bitwise operations on integer values and print the results. If not, you can skip this task.

```
a=5   b=2
print(a & b)          #output 0
print(a | b)          #output 7
print(a ^ b)          #output 7
print(~a)             #output -6
print(~b)             #output -3
print(a<<1)           #output 10
print(a>>1)           #output 2
```

26: Identity and Membership Operators

->Create a list my_list containing a few elements.

Ans: —----------

```
# Creating a list with some elements
my_list = [1, 2, 3, 4, 5]

# Display the list
print("my_list contains:", my_list)
```

->Use identity operators (is and is not) to check if two variables are the same object.

Ans:

```
# Creating two variables
a = [1, 2, 3]
b = a                    # b points to the same list object as a
c = [1, 2, 3]            # c is a new list with the same content, but different objects

# Using the 'is' operator
print("a is b:", a is b)  output: True (because a and b point to the same object)
```

Python Assignment

 # Using the 'is not' operator

print("a is not c:", a is not c) o/p:True, (because a and c point to differ object)

 # Using 'is' to check for identity with a new object

print("b is c:", b is c) o/p: False, (because b and c are different objects)

->Use membership operators (in and not in) to check if an element is present in my_list.

```
# Creating a list
my_list = [10, 20, 30, 40, 50]
```

 # Using 'in' to check if an element is in the list

- **element1 = 30**
- **element2 = 100**
- **Print(element1 in my_list) o/p: True, 30 is in the list**
- **print(element2 in my_list) o/p: False, 100 is not in the list**

 # Using 'not in' to check if an element is NOT in the list

- **print(element1 not in my_list) o/p: False, 30 is in the list**
- **print(element2 not in my_list) o/p: True, 100 is not in the list**