



Parul[®]
University

FACULTY OF ENGINEERING AND TECHNOLOGY
BACHELOR OF TECHNOLOGY

PYTHON FULL STACK DEVELOPMENT

(303105257)

IV SEMESTER

Computer Science & Engineering Department

Laboratory Manual

Academic Year: 2023-24

Certificate

This is to certify that Ms/Mr _____ with
enrollment no. _____ has successfully
completed his/her laboratory programs in the subject with Code
_____ from the department of
_____ during the academic year _____.



Date of submission

Faculty in charge

Head of the Department

INDEX

Sr No.	Date Of Performance	Aim	Date Of Submission	Signature	Marks
SET-1					
1	28/11/23	Temperature conversion			
2	28/11/23	Area and Perimeter of rectangle			
3	5/12/23	Random Password generation			
4	5/12/23	Average of a List			
5	12/12/23	Leap Year validation			
6	12/12/23	Factorial of a number			
7	19/12/23	Check string if Palindrome			
8	19/12/23	Sorting a List			
9	26/12/23	Multiplication table of a number			
10	26/12/23	Base Conversion of a number			

PRACTICAL-1

- ❖ AIM : A program that converts temperatures from Fahrenheit to Celsius and vice versa.

Problem Statement:

Develop a program that allows users to convert temperatures between Fahrenheit and Celsius. The program should provide a user-friendly interface for temperature conversion and accurately perform the calculations.

Program Description:

Create a temperature conversion program that takes input from the user, specifying the temperature in either Fahrenheit or Celsius, and outputs the equivalent temperature in the other unit. The program should handle both Fahrenheit to Celsius and Celsius to Fahrenheit conversions.

Algorithm:

1. Start
2. Display a menu to the user with options for temperature conversion.
3. Accept user input for the temperature value and the unit (Fahrenheit or Celsius).
4. Based on the user's selection, use the appropriate formula to convert the temperature.
 - o Fahrenheit to Celsius: $C=5/9 \times (F-32)$
 - o Celsius to Fahrenheit: $F=59 \times C+32$

$$C=5/9 \times (F-32)$$

$$\circ \text{ Celsius to Fahrenheit: } F=59 \times C+32$$

$$F=9/5 \times C+32$$

5. Display the converted temperature.
6. End

Formulae:

- Fahrenheit to Celsius: $C=5/9 \times (F-32)$
- Celsius to Fahrenheit: $F=59 \times C+32$

Source Code:

```
while True:  
    ch=int(input("Enter your choice:\n1.Fahrenheit\n2.Celsius\n3.Exit\n"))  
    if ch == 1:  
        far= float(input("\nEnter the temperature in Fahrenheit: "))  
        cel= (far - 32) * (5/9)  
        print("\nThe temperature in Celsius is: {:.2f}°C".format(cel))  
    elif ch == 2:  
        cel = float(input("\nEnter the temperature in Celsius: "))  
        far = (cel * 9/5) + 32  
        print("\nThe temperature in Fahrenheit is: {:.2f}°F".format(far))  
    elif ch == 3:  
        break  
    else:  
        print("\nInvalid choice. Please enter 1, 2, or 3.")
```

Expected Output:

STDIN

```
1  
183  
2  
67  
3
```

Output:

```
Enter your choice:  
1.Fahrenheit  
2.Celsius  
3.Exit  
  
Enter the temperature in Fahrenheit:  
The temperature in Celsius is: 83.89°C  
Enter your choice:  
1.Fahrenheit  
2.Celsius  
3.Exit  
  
Enter the temperature in Celsius:  
The temperature in Fahrenheit is: 152.60°F  
Enter your choice:  
1.Fahrenheit  
2.Celsius  
3.Exit
```

Actual Output:

STDIN

1
183
2
67
3

Output:

Enter your choice:
1.Fahrenheit
2.Celsius
3.Exit

Enter the temperature in Fahrenheit:
The temperature in Celsius is: 83.89°C
Enter your choice:
1.Fahrenheit
2.Celsius
3.Exit

Enter the temperature in Celsius:
The temperature in Fahrenheit is: 152.60°F
Enter your choice:
1.Fahrenheit
2.Celsius
3.Exit

Result:

The program successfully converts temperatures between Fahrenheit and Celsius as expected. Users can easily input a temperature value and select the desired conversion type, and the program provides accurate results based on the chosen formula. The practical achieves its aim of creating a user-friendly temperature conversion tool.

PRACTICAL- 2

- ❖ AIM:  A program that calculates the area and perimeter of a rectangle.

Problem Statement:

Develop a program that allows users to calculate the area and perimeter of a rectangle. The program should provide a user-friendly interface for these calculations.

Program Description:

Create a rectangle calculator program that prompts the user to choose between calculating the area or perimeter of a rectangle. The program should accept user inputs for the length and width of the rectangle, perform the calculations based on the user's choice, and display the results.

Algorithm:

1. Start
2. Display a menu to the user with options for calculation (Area, Perimeter, Exit).
3. Accept user input for the choice.
4. If the choice is 1, accept user inputs for the length and width, calculate and display the area.
5. If the choice is 2, accept user inputs for the length and width, calculate and display the perimeter.
6. If the choice is 3, exit the program.
7. Repeat the process until the user chooses to exit.
8. End

Formulae:

- Area of Rectangle of length 'l' and breadth 'b': $A = l \times b$
- Perimeter of Rectangle of length 'l' and breadth 'b': $P = 2 \times (l + b)$

Source Code:

```
while True:  
    l =float(input("Enter the length: "))  
    b = float(input("\nEnter the breadth: "))  
    print("\n1.Area\n2.Perimeter\n3.Exit")  
    cho=int(input("\nEnter the choice: "))  
    if cho == 1:  
        area = l*b  
        print("\nArea of the rectangle with length",end=" ")  
        print(l,"and breadth",b,"is: ",area)  
    elif cho==2:  
        per = 2*(l+b)  
    elif cho==3:  
        break  
    else:  
        print("Invalid Input!")
```

Expected Output:

STDIN

20.5
10.5
1
6
5
2



Output:

Enter the length:
Enter the breadth:
1.Area
2.Perimeter
3.Exit

Enter the choice:
Area of the rectangle with length 20.5 and breadth 10.5 is: 215
Enter the length:
Enter the breadth:
1.Area
2.Perimeter
3.Exit

Actual Output:

STDIN

```
20.5  
10.5  
1  
6  
5  
2
```



Output:

Enter the length:

Enter the breadth:

- 1.Area
- 2.Perimeter
- 3.Exit

Enter the choice:

Area of the rectangle with length 20.5 and breadth 10.5 is: 215

Enter the length:

Enter the breadth:

- 1.Area
- 2.Perimeter
- 3.Exit

Result:

The program successfully calculates the area and perimeter of a rectangle based on the user's choice. It provides a user-friendly interface and accurate results for the calculations, meeting the goal of creating a rectangle calculator.

PRACTICAL- 3

- ❖ AIM: A program that generates a random password of a specified length.

Problem Statement:

Develop a program that generates a random password of a specified length. The program should allow users to input the desired length and provide a secure and random password as output.

Program Description:

Create a password generator program that takes user input for the desired length of the password. The program should use a combination of uppercase letters, lowercase letters, digits, and special characters to create a random and secure password.

Algorithm:

1. Start
2. Accept user input for the desired length of the password.
3. Check if the input is a positive integer; if not, ask the user to enter a valid length.
4. Generate a random password using a combination of uppercase letters, lowercase letters, digits, and special characters.
5. Display the generated password.
6. End

Source Code:

```
import random
import string

size = int(input("Enter the size of password: "))
allchar = string.ascii_letters+string.digits+string.punctuation
password = ''.join(random.choice(allchar) for i in range(size))
print(password)
```

Expected Output:

STDIN

10

Output:

Enter the size of password: ?d\p!w=1!4

Actual Output:

STDIN

10

Output:

Enter the size of password: ?d\p!w=1!4

Result:

The program successfully generates a random password of the specified length, providing users with a secure password for various applications. The password includes a mix of uppercase letters, lowercase letters, digits, and special characters, ensuring randomness and complexity.

PRACTICAL- 4

- ❖ AIM: A program that calculates the average of a list of numbers.

Problem Statement:

Develop a program that calculates the average of a list of numbers. The program should take a list of numbers as input, perform the calculation, and display the average.

Program Description:

Create an average calculator program that prompts the user to input a list of numbers. The program should then calculate the average of the provided numbers and display the result.

Algorithm:

1. Start
2. Accept user input for a list of numbers.
3. Convert the input into a list of numeric values.
4. Calculate the sum of the numbers in the list.
5. Calculate the average by dividing the sum by the total number of elements in the list.
6. Display the calculated average.
7. End

Formulae:

- Average = sum of all the elements / number of elements.

Source Code:

```
while True:  
    print("1. Calculate the Average of the list.\n2.Exit\n")  
    cho = int(input("Enter your choice: "))  
    if cho == 1:  
        lis = eval(input("\nEnter list of elements: "))  
        average = sum(lis)/len(lis)  
        print("\nAverage of the list =",average)  
    elif cho==2:  
        break  
    else:  
        print("\nInvalid Input!")
```

Expected Output:

STDIN

```
1  
[2,3,12,4,2,33,2]  
2
```



Output:

1. Calculate the Average of the list.
2. Exit

Enter your choice:

Enter list of elements:

Average of the list = 8.285714285714286

1. Calculate the Average of the list.
2. Exit

Enter your choice:

Actual Output:

STDIN

```
1  
[2,3,12,4,2,33,2]  
2
```



Output:

1. Calculate the Average of the list.
2. Exit

Enter your choice:

Enter list of elements:

Average of the list = 8.285714285714286

1. Calculate the Average of the list.
2. Exit

Enter your choice:

Result:

The program successfully calculates the average of a list of numbers provided by the user. It handles the input, performs the necessary calculations, and displays the average. The result is achieved as intended, providing users with an average value for the input list.

PRACTICAL- 5

- ❖ AIM: A program that checks if a given year is a leap year.

Problem Statement:

Develop a program that checks if a given year is a leap year. The program should take a year as input, perform the leap year check, and display whether the given year is a leap year or not.

Program Description:

Create a leap year checker program that prompts the user to input a year. The program should then determine whether the provided year is a leap year according to the leap year rules.

Algorithm:

1. Start
2. Accept user input for the year.
3. Check if the year is evenly divisible by 4.
4. If it is, check if the year is divisible by 100.
5. If it is, check if the year is divisible by 400.
6. If any of the above conditions are true, the year is a leap year; otherwise, it is not.
7. Display the result.
8. End

Source Code:

```
def is_leap_year(year):
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False

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

if input_year <= 0:
    raise ValueError("Please enter a valid positive year.")

if is_leap_year(input_year):
    print("{} is a leap year.".format(input_year))
else:
    print("{} is not a leap year.".format(input_year))
```

Expected Output:

STDIN

2022

Output:

Enter a year:

2022 is not a leap year.

Actual Output:

STDIN

2022

Output:

Enter a year:

2022 is not a leap year.

Result:

The program successfully checks whether a given year is a leap year or not. It follows the leap year rules and provides the user with a clear indication of the leap year status for the provided year.

PRACTICAL- 6

- ❖ AIM: A program that calculates the factorial of a number.

Problem Statement:

Develop a program that calculates the factorial of a given number. The program should take a positive integer as input, perform the factorial calculation, and display the result.

Program Description:

Create a factorial calculator program that prompts the user to input a positive integer. The program should then calculate the factorial of the provided number.

Algorithm:

1. Start
2. Accept user input for a positive integer.
3. Check if the input is a positive integer; if not, ask the user to enter a valid positive integer.
4. Calculate the factorial of the input number using a loop.
5. Display the calculated factorial.
6. End

Source Code:

```
n = int(input("Enter a number: "))

factorial = 1

if n<0:

    print("Factorial does not exist for negative numbers.")

elif n==0:

    print("The factorial of 0 is 1.")

else:

    for i in range(1,n+1):

        factorial*=i

    print("The factorial of",n,"is",factorial)
```

Expected Output:

STDIN

10

Output:

Enter a number:

The factorial of 10 is 3628800

Actual Output:

STDIN

10

Output:

Enter a number:

The factorial of 10 is 3628800

Result:

The program successfully calculates the factorial of a given positive integer. It handles input validation, performs the necessary calculations using a loop, and displays the calculated factorial. Users can obtain the factorial of the provided number with this program.

PRACTICAL- 7

❖ AIM: A program that checks if a given string is a palindrome.

Problem Statement:

Develop a program that checks if a given string is a palindrome. The program should take a string as input, perform the palindrome check, and display whether the given string is a palindrome or not.

Program Description:

Create a palindrome checker program that prompts the user to input a string. The program should then determine whether the provided string is a palindrome.

Algorithm:

1. Start
2. Accept user input for the string.
3. Remove spaces and convert the string to lowercase for case-insensitive comparison.
4. Compare the string with its reverse.
5. If the string is the same when read backward, it is a palindrome; otherwise, it is not.
6. Display the result.
7. End

Source Code:

```
my_str = input("Enter string: ")

my_str = my_str.casefold()

rev_str = my_str[::-1]

if my_str == rev_str:

    print("The string is a palindrome string.")

else:

    print("The string is not a palindrome string.")
```

Expected Output:

STDIN

radar

Output:

Enter string:

The string is a palindrome string.

Actual Output:

STDIN

radar

Output:

Enter string:

The string is a palindrome string.

Result:

The program successfully checks whether a given string is a palindrome or not. It handles input cleaning, performs the palindrome check, and provides the user with a clear indication of whether the provided string is a palindrome.

PRACTICAL – 8

- ❖ AIM: A program that sorts a list of numbers in ascending or descending order.

Problem Statement:

Develop a program that sorts a list of numbers in either ascending or descending order. The program should take a list of numbers and a sorting order as input, perform the sorting, and display the sorted list.

Program Description:

Create a list sorting program that prompts the user to input a list of numbers and the desired sorting order. The program should then sort the list accordingly.

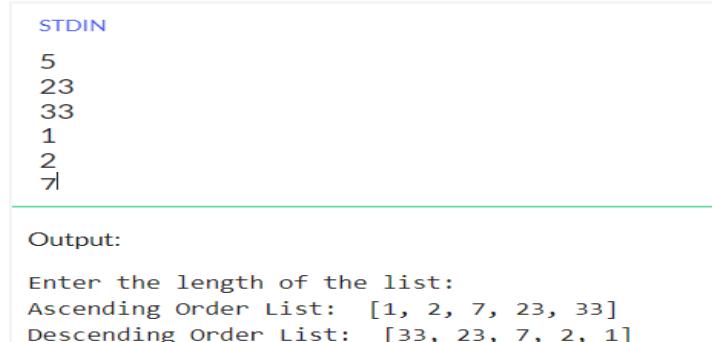
Algorithm:

1. Start
2. The list is provided.
3. Sort the list in ascending or descending order based on the user's choice.
4. Display the sorted list.
5. End

Source Code:

```
def sort_numbers(numbers, reverse=False):
    return sorted(numbers, reverse=reverse)
my_list = [5,2,9,7,4,1]
print("Ascending Order List: ",sort_numbers(my_list,reverse=False))
print("Descending Order List: ",sort_numbers(my_list,reverse=True))
```

Expected Output:

STDIN	<pre>5 23 33 1 2 7</pre>
	
Output:	<pre>Enter the length of the list: Ascending Order List: [1, 2, 7, 23, 33] Descending Order List: [33, 23, 7, 2, 1]</pre>

Actual Output:

STDIN

```
5  
23  
33  
1  
2  
7|
```



Output:

```
Enter the length of the list:  
Ascending Order List: [1, 2, 7, 23, 33]  
Descending Order List: [33, 23, 7, 2, 1]
```

Result:

The program successfully sorts a list of numbers in either ascending or descending order based on user input. It handles input validation, performs the sorting, and displays the sorted list accordingly. Users can obtain the sorted list with this program.

PRACTICAL- 9

- ❖ AIM: A program that generates a multiplication table for a given number.

Problem Statement:

Develop a program that generates a multiplication table for a given number. The program should take a positive integer as input, perform the multiplication table generation, and display the result.

Program Description:

Create a multiplication table generator program that prompts the user to input a positive integer. The program should then generate and display the multiplication table for the provided number.

Algorithm:

1. Start
2. Accept user input for a positive integer.
3. Check if the input is a positive integer; if not, ask the user to enter a valid positive integer.
4. Generate the multiplication table for the input number.
5. Display the multiplication table.
6. End

Source Code:

```
n = int(input("Enter a number: "))

for i in range(1,11):

    print(f"{n} x {i} = {n*i}")
```

Expected Output:

```
STDIN  
20  
  
Output:  
Enter a number: 20 x 1 = 20  
20 x 2 = 40  
20 x 3 = 60  
20 x 4 = 80  
20 x 5 = 100  
20 x 6 = 120  
20 x 7 = 140  
20 x 8 = 160  
20 x 9 = 180  
20 x 10 = 200
```

Actual Output:

```
STDIN  
20  
  
Output:  
Enter a number: 20 x 1 = 20  
20 x 2 = 40  
20 x 3 = 60  
20 x 4 = 80  
20 x 5 = 100  
20 x 6 = 120  
20 x 7 = 140  
20 x 8 = 160  
20 x 9 = 180  
20 x 10 = 200
```

Result:

The program successfully generates and displays the multiplication table for a given positive integer. It handles input validation and provides the user with a clear and formatted multiplication table for the specified number. Users can obtain the multiplication table with this program.

PRACTICAL – 10

- ❖ AIM: A program that converts a given number from one base to another

Problem Statement:

Develop a program that converts a given number from one base to another. The program should take a number, its base, and the target base as input, perform the conversion, and display the result.

Program Description:

Create a number base conversion program that prompts the user to input a number, its base, and the target base. The program should then convert the given number from its base to the target base and display the result.

Algorithm:

1. Print a menu for the user to choose the source number system (**decimal, binary, octal, or hexadecimal**).
2. Take user input for the source number system (**input_choice**).
3. Take additional input for the number to be converted based on the source system.
4. Print a menu for the user to choose the target conversion system.
5. Take user input for the target conversion (**choice**).
6. Call the appropriate conversion function based on the source system and target conversion.
7. Print the result of the conversion.

Source Code:

```
def decimal_others(value,choice):  
  
    if choice==1:  
        return value  
  
    elif choice==2:  
        return '{0:b}'.format(value)  
  
    elif choice==3:  
        return '{0:o}'.format(value)  
  
    elif choice==4:
```

```
        return '{0:x}'.format(value)

    else:

        return "Invalid Option"

def binary_others(value,choice):

    if choice==1:

        return value

    elif choice==2:

        return int(value,2)

    elif choice==3:

        return '{0:o}'.format(int(value,2))

    elif choice==4:

        return '{0:x}'.format(int(value,2))

    else:

        return "Invalid Option"

def octal_others(value,choice):

    if choice==1:

        return value

    elif choice==2:

        return int(value,8)

    elif choice==3:

        return '{0:b}'.format(int(value,8))

    elif choice==4:

        return '{0:x}'.format(int(value,8))

    else:

        return "Invalid Option"

def hex_others(value,choice):

    if choice==1:
```

```

        return value

    elif choice==2:

        return int(value,16)

    elif choice==3:

        return '{0:0}'.format(int(value,16))

    elif choice==4:

        return '{0:b}'.format(int(value,16))

    else:

        return "Invalid Option"

print("Convert from: 1: decimal ,2: binary,3: octal 4:hexadecimal")

input_choice=int(input("Enter the choice"))

if input_choice==1:

    decimal_num=int(input("Enter decimal number"))

    print('Convert to: 1: decimal ,2: binary,3: octal 4:hexadecimal')

    choice=int(input("Enter Target conversion:\n"))

    print("Converted value: ",decimal_others(decimal_num,choice))

elif input_choice==2:

    binary_num=input("Enter decimal number")

    print('Convert to: 1: binary ,2: decimal,3: octal 4:hexadecimal')

    choice=int(input("Enter Target conversion:\n"))

    print("Converted value: ",binary_others(binary_num,choice))

elif input_choice==3:

    octal_num=input("Enter decimal number")

    print('Convert to: 1: octal ,2: decimal,3: binary 4:hexadecimal')

    choice=int(input("Enter Target conversion:\n"))

    print("Converted value: ",octal_others(octal_num,choice))

```

```
elif input_choice==4:  
  
    hex_num=input("Enter decimal number")  
  
    print('Convert to: 1: hex,2: decimal,3: octal 4:binary')  
  
    choice=int(input("Enter Target conversion:\n"))  
  
    print("Converted value: ",hex_others(hex_num,choice))
```

Expected Output:

STDIN

```
1|  
23  
2
```

Output:

Convert from:

```
1: decimal  
2: binary  
3: octal  
4: hexadecimal  
5:Exit
```

Enter the choice

Enter decimal numberConvert to:

```
1: decimal ,  
2: binary  
3: octal  
4: hexadecimal
```

Enter Target conversion:

Converted value: 10111

Convert from:

```
1: decimal  
2: binary  
3: octal  
4: hexadecimal  
5:Exit
```

Actual Output:

STDIN

```
1|  
23  
2
```

Output:

Convert from:

1: decimal
2: binary
3: octal
4: hexadecimal
5: Exit

Enter the choice

Enter decimal numberConvert to:

1: decimal ,
2: binary
3: octal
4: hexadecimal

Enter Target conversion:

Converted value: 10111

Convert from:

1: decimal
2: binary
3: octal
4: hexadecimal
5: Exit

Result:

The program successfully converts a given number from one base to another. It prompts the user for the input number, the base of the given number, and the target base. It then performs the conversion and displays the result. Users can use this program to convert numbers between different bases.