**DR. BR Ambedkar National Institute of**

**Technology, Jalandhar**



**Center of Excellence AI (Session 2025-27)**

Programming with Python Laboratory : AI-523

**Submitted to:**

Dr**.** Diksha Kumari Dr. Ruchika Arora

**Submitted by:**

Rohit Kumar

Roll No: 25901334

M.Tech AI (2025-27)

**CONTENTS:**

|  |  |  |
| --- | --- | --- |
| **Experiment Name** | **Date** | **Page No** |
| 1. Installation and Basic Programs (Variables and Data Types, Arithmetic, Comparison, Assignment, Logical Operators) | 28/08/25 | 1 |
| 2. Operators based programs | 04/09/25 | 2 |
| 3. String Manipulation, Number System and Conversions | 11/09/25 | 2 |
| 4. Operator Precedence, Conditional Statements, Loops (For and While) | 18/09/25 | 3 |
| 5. Nested Loops, Keyword Arguments | 25/09/25 | 4 |
| 6. Program based on Functions | 26/09/25 | 6 |
| 7. Modules and packages | 30/9/25 | 7 |
| 8. String and its operations | 16/10/25 | 8 |
| 9. File Handling | 30/10/25 | 9 |
| 10. Data Structure: Lists, Tuples, Sets | 06/11/25 | 10 |
| 11. Programming Exercises with classes and objects, Inheritance | 13/11/25 | 11 |
| 12. Operator Overloading, Error Handling | 20/11/25 | 14 |



<https://github.com/Ohi-Me/NITJ_AI/tree/main/PPL_LAB>****



In [1]:

a **=** 10

b **=** 3.14

c **=** "Python"

d **= True**

print(a, type(a))

print(b, type(b))

print(c, type(c))

print(d, type(d))

In [2]:

s **=** "Hello"

s **+=** " World"

print(s)

10 <class 'int'>

3.14 <class 'float'> Python <class 'str'> True <class 'bool'>





Hello World



In [3]:

s **=** "Hi"

print(s **+** " Python")

print(s **\*** 3)

Hi Python HiHiHi











In [4]:

**import** math *# Importing math module*

a, b, c **=** 3, 4, 5

s **=** (a **+** b **+** c) **/** 2

*# Area calculation*

area **=** math**.**sqrt(s**\***(s**-**a)**\***(s**-**b)**\***(s**-**c)) print("Area of triangle:", area)

Area of triangle: 6.0

In [ ]:

Explanation:

Semi**-**perimeter **is** calculated first**.**

Heron’s formula **is** applied using math**.**sqrt()**.**



In [6]:

*# Taking number of coins*

rs10 **=** int(input("enter no of 10 rs coin"))

rs5 **=** int(input("enter no of 5 rs coin")) rs2 **=** int(input("enter no of 2 rs coin")) rs1 **=** int(input("enter no of 1 rs coin")) *# Calculating total amount*

total **=** rs10**\***10 **+** rs5**\***5 **+** rs2**\***2 **+** rs1 print("Total Amount:", total)

Total Amount: 105





In [7]:

*# Taking decimal input*

n **=** int(input("Enter number: "))

*# Conversions*

print("Binary:", bin(n))

print("Octal:", oct(n))

print("Hexadecimal:", hex(n))

Binary: 0b11001 Octal: 0o31 Hexadecimal: 0x19







In [8]:

*# Taking float as string*

s **=** input("Enter float value: ")

*# Converting and adding*

num **=** float(s)

print(num **+** 10.5)

67.37





In [9]:

*# Step 1: Calculate power*

step1 **=** 3 **\*\*** 2

print("Step 1: 3 \*\* 2 =", step1)

*# Step 2: Multiply*

step2 **=** 2 **\*** step1

print("Step 2: 2 \* 3 \*\* 2 =", step2)

*# Step 3: Divide*

step3 **=** 8 **/** 4

print("Step 3: 8 / 4 =", step3)

*# Step 4: Final calculation*

x **=** 10 **+** step2 **-** step3

print("Step 4: 10 + 2 \* 3 \*\* 2 - 8 / 4 =", x)

In [11]:

*# Dictionary creation*

students **=** {"Amit":85, "Riya":90, "Neha":88}

print(students)

Step 1: 3 \*\* 2 = 9

Step 2: 2 \* 3 \*\* 2 = 18

Step 3: 8 / 4 = 2.0

Step 4: 10 + 2 \* 3 \*\* 2 - 8 / 4 = 26.0





*# Searching marks*

name **=** input("Enter name: ")

print(students**.**get(name, "Student not found"))

{'Amit': 85, 'Riya': 90, 'Neha': 88}

85



In [12]:

n **=** int(input("enter a number"))

rev **=** 0

*# Loop to reverse digits*

**while** n **>** 0:

rev **=** rev**\***10 **+** n**%10** n **//=** 10

print(rev)

6879876





In [13]:

n **=** int(input("Enter a number"))

*# Printing table*

**for** i **in** range(1,11):

print(n, "x", i, "=", n**\***i)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 25 | x | 1 | = | 25 |
| 25 | x | 2 | = | 50 |
| 25 | x | 3 | = | 75 |
| 25 | x | 4 | = | 100 |
| 25 | x | 5 | = | 125 |
| 25 | x | 6 | = | 150 |
| 25 | x | 7 | = | 175 |
| 25 | x | 8 | = | 200 |
| 25 | x | 9 | = | 225 |
| 25 | x | 10 = 250 | | |





In [15]:

**for** i **in** range(1, 6):

*# print spaces*

**for** s **in** range(5 **-** i): print(" ", end**=**"")

*# print numbers*

**for** j **in** range(1, i **+** 1):

print(j, end**=**"")

*# move to next line*

print()

In [16]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 45 | x | 1 | = | 45 |
| 45 | x | 2 | = | 90 |
| 45 | x | 3 | = | 135 |
| 45 | x | 4 | = | 180 |
| 45 | x | 5 | = | 225 |
| 45 | x | 6 | = | 270 |
| 45 | x | 7 | = | 315 |
| 45 | x | 8 | = | 360 |
| 45 | x | 9 | = | 405 |
| 45 | x | 10 = 450 | | |

In [17]:

**def** student\_info(name, age, course): print(name, age, course)

*# Calling with keywords*

student\_info(course**=**"AI", name**=**"Ravi", age**=**20)

1

12

123

1234

12345





n **=** int(input("Enter a number"))

*# Printing table*

**for** i **in** range(1,11):

print(n, "x", i, "=", n**\***i)





Ravi 20 AI



In [19]:

**import** math

n **=** int(input("enter a no"))

print(abs(n))

print(math**.**sqrt(n)) print(n**\*\***3)

55

7.416198487095663

166375





In [22]:

*# Lambda function*

n**=**int(input("enter a no"))

square **= lambda** x: x**\***x print(square(n))

16





In [23]:

**def** count\_digits(n):

**if** n **==** 0:

**return** 0

**return** 1 **+** count\_digits(n**//**10)

print(count\_digits(1234))

4





In [24]:

n **=** int(input("Enter number of elements: "))

lst **=** []

*# Taking input into list*

**for** i **in** range(n):

num **=** int(input("Enter element: "))

lst**.**append(num)

*# Printing original list*

print("Original list:", lst)

print("Reversed list:", end**=**" ")

**for** i **in** range(len(lst) **-** 1, **-**1, **-**1):

print(lst[i], end**=**" ")

In [25]:

**import** random **as** rnd

**for** i **in** range(5):

print(rnd**.**randint(1,100))

Original list: [45, 88, 23, 776, 67]

Reversed list: 67 776 23 88 45





96

41

5

32

27





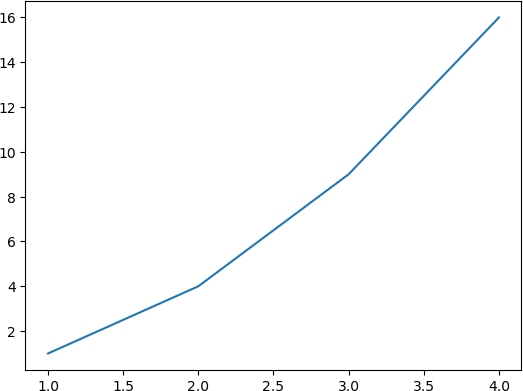
In [26]:

**import** matplotlib.pyplot **as** plt

x **=** [1,2,3,4]

y **=** [1,4,9,16]

plt**.**plot(x,y) plt**.**show()







In [27]:

s **=** input("Enter a string: ")

result **=** ""

**for** ch **in** s:

**if** ch **not in** "aeiouAEIOU":

result **=** result **+** ch

print("String without vowels:", result)

String without vowels: pythn s fn





In [28]:

s **=** input("Enter a string: ")

clean **=** ""

**for** ch **in** s:

**if** ch**.**isalnum():

*# keep only letters and numbers*

clean **+=** ch**.**lower()

**if** clean **==** clean[::**-**1]: print("Palindrome")

**else**:

print("Not a palindrome")

Not a palindrome





In [29]:

sentence **=** input("Enter a sentence: ")

words **=** sentence**.**split() *# split sentence into words*

rev\_words **=** []

**for** i **in** range(len(words) **-** 1, **-**1, **-**1):

rev\_words**.**append(words[i])

result **=** " "**.**join(rev\_words)

print("Reversed sentence:", result)

Reversed sentence: fun is python





In [32]:

f **=** open("file.txt","r") text **=** f**.**read()

print(text**.**count(" ")," whitespaces")

print(text**.**count("\n")," newline characters")

print(text**.**count("\t")," tabs")

5 whitespaces

3 newline characters

3 tabs







In [34]:

**with** open("source.bin", "wb") **as** f:

f**.**write(b"ABCDEFGHIJKLMNOPQRSTUVWXYZ")

*# Copy first 10 bytes*

**with** open("source.bin", "rb") **as** src: data **=** src**.**read(10)

**with** open("dest.bin", "wb") **as** dest: dest**.**write(data)

*# Read and print content of new file*

**with** open("dest.bin", "rb") **as** dest: content **=** dest**.**read()

print("Content of new file:", content)

Content of new file: b'ABCDEFGHIJ'





In [36]:

n **=** int(input("Enter number of elements: "))

lst **=** []

*# i) Take n integers from user*

**for** i **in** range(n):

num **=** int(input("Enter element: "))

lst**.**append(num)

*# ii) Create new list without duplicates*

new\_list **=** []

**for** x **in** lst:

**if** x **not in** new\_list: new\_list**.**append(x)

*# iii) Print both lists*

print("Original list:", lst)

print("List without duplicates:", new\_list)

Original list: [55, 77, 55, 77, 34]

List without duplicates: [55, 77, 34]



b) Write a Python program to: Create a tuple, e.g. nums = (10, 20, 30, 40, 50). Ask the user to enter a number. Print the index of that number in the tuple if it exists, otherwise print a message saying it is not present.

nums **=** (10, 20, 30, 40, 50)

n **=** int(input("Enter a number to search: "))

**if** n **in** nums:

print("Number found at index:", nums**.**index(n))

**else**:

print("Number is not present in the tuple")

In [37]:

Number found at index: 3





In [38]:

*# Read first sequence*

n1 **=** int(input("Enter number of elements in first sequence: "))

seq1 **=** []

**for** i **in** range(n1):

seq1**.**append(int(input("Enter element: ")))

*# Read second sequence*

n2 **=** int(input("Enter number of elements in second sequence: "))

seq2 **=** []

**for** i **in** range(n2):

seq2**.**append(int(input("Enter element: ")))

*# Convert to sets to remove duplicates*

set1 **=** set(seq1) set2 **=** set(seq2)

*# Common elements*

common **=** sorted(set1 **&** set2)

*# Elements in exactly one sequence (symmetric difference)*

sym\_diff **=** sorted(set1 **^** set2)

*# Print results*

print("Common elements:", common)

print("Symmetric difference:", sym\_diff)

Common elements: []

Symmetric difference: [22, 23, 34, 55, 67, 75, 79, 88]

Explanation: This program reads two integer sequences, converts them into

**and** then prints the sorted common elements **and** the sorted symmetric difference

In [ ]:









In [39]:

**class** Employee:

count **=** 0 *# class variable to track number of employees*

**def**  init (self, name, designation, salary): self**.**name **=** name

self**.**designation **=** designation self**.**salary **=** salary

Employee**.**count **+=** 1

**def** display(self):

print("Name:", self**.**name)

print("Designation:", self**.**designation) print("Salary:", self**.**salary)

print()

*# Creating employee objects*

e1 **=** Employee("Amit", "Manager", 50000)

e2 **=** Employee("Riya", "Developer", 40000)

e3 **=** Employee("Neha", "Tester", 35000)

*# Display details*

e1**.**display() e2**.**display() e3**.**display()

*# Display total employees*

print("Total number of employees:", Employee**.**count)

Name: Amit Designation: Manager Salary: 50000

Name: Riya Designation: Developer Salary: 40000

Name: Neha Designation: Tester Salary: 35000

Total number of employees: 3









In [40]:

*# Department class*

**class** Department:

**def**  init (self, name): self**.**name **=** name

*# Course class*

**class** Course:

**def**  init (self, name, year, department): self**.**name **=** name

self**.**year **=** year

self**.**department **=** department

*# Student class*

**class** Student:

**def**  init (self, name, rollno, year): self**.**name **=** name

self**.**rollno **=** rollno self**.**year **=** year

self**.**course **= None**

**def** enroll(self, course): self**.**course **=** course

**def** display(self):

print("Student Name:", self**.**name) print("Roll No:", self**.**rollno)

print("Year:", self**.**year)

print("Course:", self**.**course**.**name)

print("Department:", self**.**course**.**department**.**name)

*# Creating objects*

dept **=** Department("Computer Science")

course **=** Course("Python Programming", 2025, dept) student **=** Student("Riya", 101, 2025)

*# Enroll student*

student**.**enroll(course)

*# Display details*

student**.**display()

Student Name: Riya Roll No: 101

Year: 2025

Course: Python Programming Department: Computer Science









In [41]:

**class** Student:

**def**  init (self, name, marks): self**.**name **=** name

self**.**marks **=** marks

**def**  add (self, other):

**return** self**.**marks **+** other**.**marks

*# Creating student objects*

s1 **=** Student("Amit", 85)

s2 **=** Student("Riya", 90)

*# Using overloaded + operator*

total\_marks **=** s1 **+** s2

print("Total Marks:", total\_marks)

Total Marks: 175





In [43]:

name **=** input("Enter your name: ")

age **=** int(input("Enter your age: "))

*# Validate name*

**if not** name**.**isalpha():

print("Invalid name entered")

**else**:

*# Validate age for voting*

**if** age **>=** 18:

print(name, "is eligible to vote")

**else**:

print(name, "is not eligible to vote")

riya is eligible to vote





In [44]:

**while True**: **try**:

n **=** int(input("Enter an integer: "))

print("You entered:", n)

**break**

**except** ValueError:

print("Invalid input, please enter an integer.")

Invalid input, please enter an integer. Invalid input, please enter an integer. You entered: 45



