

INDEX

<u>SERIAL NO.</u>	<u>TOPIC</u>	<u>PAGE NO.</u>
1.	PYTHON PROGRAMS	
2.	GRAPH PROGRAMS	
3.	CAPSTONE PROJECT: THE ETHICS MORAL EDUCATION	

P2.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P2.py (3.13.5)

File Edit Format Run Options Window Help

```
name = input("Enter your name: ")
age = input("Enter your age: ")
print(name, "your age is", age)
```

PROGRAM 1

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P2.py =====
=====
Enter your name: Tejas
Enter your age: 15
Tejas your age is 15
```

P3.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P3.py (3.13.5)

File Edit Format Run Options Window Help

```
# This is a single line comment
print("Hello, world!") # This is another single line comment
```

Program 2

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P3.py =====
=====
Hello, world!
```

P4.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P4.py (3.13.5)

File Edit Format Run Options Window Help

```
This program demonstrates basic arithmetic operations without using functions.
It performs addition, subtraction, multiplication, and division of two numbers.
...
# Uses of Arithmetic Operators
# Addition
a = 10
b = 5
result_add = a + b
print("Addition:", result_add)

# Subtraction
result_sub = a - b
print("Subtraction:", result_sub)

# Multiplication
result_mul = a * b
print("Multiplication:", result_mul)

# Division
result_div = a / b
print("Division:", result_div)
```

Program 3

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P4.py =====
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0
```

P5.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P5.py (3.13.5)

File Edit Format Run Options Window Help

```
# The input() function returns a string
# So we use the int() function to convert the strings to integers

num1 = int(input("Enter first number"))
num2 = int(input("Enter second number"))

# Perform arithmetic operations
addition = num1 + num2
subtraction = num1 - num2
multiplication = num1 * num2
division = num1 / num2
modulus = num1 % num2
exponentiation = num1 ** num2
floor_division = num1 // num2

# Display the results
print(num1, "+", num2, "=", addition)
print(num1, "-", num2, "=", subtraction)
print(num1, "*", num2, "=", multiplication)
print(num1, "/", num2, "=", division)
print(num1, "%", num2, "=", modulus)
print(num1, "**", num2, "=", exponentiation)
print(num1, "//", num2, "=", floor_division)
```

Program 4

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.

>>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P5.py =====
Enter first number5
Enter second number25
5 + 25 = 30
5 - 25 = -20
5 * 25 = 125
5 / 25 = 0.2
5 % 25 = 5
5 ** 25 = 298023223876953125
5 // 25 = 0
```

P6.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P6.py (3.13.5)

File Edit Format Run Options Window Help

```
# Uses of Comparison Operators

# Equal to
print("Equal to:", 10 == 5)

# Not equal to
print("Not equal to:", 10 != 5)

# Greater than
print("Greater than:", 10 > 5)

# Less than
print("Less than:", 10 < 5)

# Greater than or equal to
print("Greater than or equal to:", 10 >= 5)

# Less than or equal to
print("Less than or equal to:", 10 <= 5)
```

Program 5

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.

>>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P6.py =====
=
Equal to: False
Not equal to: True
Greater than: True
Less than: False
Greater than or equal to: True
Less than or equal to: False
...
```

P7.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P7.py (3.13.5)

File Edit Format Run Options Window Help

```
# Assignment
x = 15
print("Assignment:", x)

# Addition assignment
x += 3
print("Addition Assignment:", x)

# Subtraction assignment
x -= 2
print("Subtraction Assignment:", x)

# Multiplication assignment
x *= 4
print("Multiplication Assignment:", x)

# Division assignment
x /= 2
print("Division Assignment:", x)

# Modulus assignment
x %= 3
print("Modulus Assignment:", x)

# Floor division assignment
x //= 2
print("Floor Division Assignment:", x)

# Exponentiation assignment
x **= 3
print("Exponentiation Assignment:", x)
```

Program 6

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Enter "help" below or click "Help" above for more information.
```

```
>>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P7.py =====
Assignment: 15
Addition Assignment: 18
Subtraction Assignment: 16
Multiplication Assignment: 64
Division Assignment: 32.0
Modulus Assignment: 2.0
Floor Division Assignment: 1.0
Exponentiation Assignment: 1.0
```

P8.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P8.py (3.13.5)

File Edit Format Run Options Window Help

```
# Uses of Logical Operators
```

Program 7

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
```

```
>>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P8.py =====
Logical AND Result: True
Logical OR Result: True
Logical NOT Result: True
```

P9.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P9.py (3.13.5)

File Edit Format Run Options Window Help

```
# Uses of Bitwise Operators

# Bitwise AND
a = 10 # Binary: 1010
b = 6 # Binary: 0110
result_and = a & b
print("Bitwise AND:", result_and)

# Bitwise OR
result_or = a | b
print("Bitwise OR:", result_or)

# Bitwise XOR
result_xor = a ^ b
print("Bitwise XOR:", result_xor)

# Bitwise NOT
result_not = ~a
print("Bitwise NOT:", result_not)

# Bitwise Left Shift
result_left_shift = a << 2
print("Bitwise Left Shift:", result_left_shift)

# Bitwise Right Shift
result_right_shift = a >> 2
print("Bitwise Right Shift:", result_right_shift)
```

Program 8

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>
===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P9.py =====
Bitwise AND: 2
Bitwise OR: 14
Bitwise XOR: 12
Bitwise NOT: -11
Bitwise Left Shift: 40
Bitwise Right Shift: 2
>>
```

P10.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P10.py (3.13.5)

File Edit Format Run Options Window Help

```
# Uses of Membership Operators

# in operator
my_list = [1, 2, 3, 4, 5]
print("Is 3 in the list?", 3 in my_list)

# not in operator
print("Is 6 not in the list?", 6 not in my_list)
```

Program 9

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P10.py =====
Is 3 in the list? True
Is 6 not in the list? True
>>>
```

P11.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P11.py (3.13.5)

File Edit Format Run Options Window Help

```
# Uses of Identity Operators

# is
x = [1, 2, 3]
y = [1, 2, 3]
print("Is x the same object as y?", x is y)

# is not
print("Is x not the same object as y?", x is not y)
```

Program 10

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.

>>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P11.py =====
Is x the same object as y? False
Is x not the same object as y? True
>>>
```

P12.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P12.py (3.13.5)

File Edit Format Run Options Window Help

```
# Demonstrating type casting in Python

# Initial values of different types
int_value = 10
float_value = 15.5
str_value = "25"
bool_value = True

# Type casting
# Convert integer to float
int_to_float = float(int_value)
# Convert float to integer
float_to_int = int(float_value)
# Convert string to integer
str_to_int = int(str_value)
# Convert boolean to integer
bool_to_int = int(bool_value)

# Print the casted values and their new types
print("\nCasted values and their new types:")
print("int_to_float:", int_to_float, type(int_to_float))
print("float_to_int:", float_to_int, type(float_to_int))
print("str_to_int:", str_to_int, type(str_to_int))
print("bool_to_int:", bool_to_int, type(bool_to_int))
```

Program 11

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.

>>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P12.py =====

Casted values and their new types:
int_to_float: 10.0 <class 'float'>
float_to_int: 15 <class 'int'>
str_to_int: 25 <class 'int'>
bool_to_int: 1 <class 'int'>
...
```

P13.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P13.py (3.13.5)

File Edit Format Run Options Window Help

Perform an Addition
num1 = 10
num2 = 20

result = num1 + num2
Display the result
print("The result of", num1, "+", num2, "is:", result)

Program 12

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P13.py =====
The result of 10 + 20 is: 30
>>>

P14.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P14.py (3.13.5)

File Edit Format Run Options Window Help

x = 10

if x > 5:
 print("x is greater than 5")

Program 13

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P14.py =====
x is greater than 5
>>>

P15.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P15.py (3.13.5)

File Edit Format Run Options Window Help

Program 14

```
# Asking the user for input
temperature = float(input("Please enter the temperature in Celsius: "))

# Checking if the temperature is above or below freezing point
if temperature <= 0:
    print("It's freezing!")
else:
    print("It's not freezing.")
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.

>>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P15.py ====
Please enter the temperature in Celsius: 40
It's not freezing.
```

P16.py - C:/Users/DELL/AppData/Local/Programs/Python/Python313/P16.py (3.13.5)

File Edit Format Run Options Window Help

Program 15

```
# Asking the user for input
score = int(input("Please enter your exam score: "))

# Grading the score
if score >= 90:
    print("Your grade is A.")
elif score >= 80:
    print("Your grade is B.")
elif score >= 70:
    print("Your grade is C.")
elif score >= 60:
    print("Your grade is D.")
else:
    print("Your grade is F. You need to retake the exam.")
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)]
Enter "help" below or click "Help" above for more information.

>>> ===== RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python313/P16.py ====
Please enter your exam score: 83
Your grade is B.
```

PRACTICAL 1

PROGRAM 16:

```
#program 16:to demonstrate the use of the if-elif-else statement
#asking user for input
score = int(input("please enter your exam score : "))
#grading the score
if score >=90:
    print("your grade is A.")
elif score >=80:
    print("your grade is B.")
elif score >=70:
    print("your grade is C.")
elif score >=60:
    print("your grade is D.")
else:
    print("your grade is F. you need to retake the exam")
```

THE OUTPUT:

```
===== RESTART: C:/Users/Admin/akash verma.py =====
please enter your exam score : 77
your grade is C.
```

PROGRAM 17:

```

# Program 17: To demonstrate the use of nested if statement

# Input the total purchase amount
total_purchase_amount = float(input("Enter the total purchase amount (Rs): "))

# Initialize the discount rate and discounted amount
discount_rate = 0
discount_amount = 0

# Determine the discount rate using nested if statements
if total_purchase_amount >= 10000:
    discount_rate = 0.10 # 10% discount
    print("Congratulations! You get a 10% discount.")
else:
    if total_purchase_amount >= 5000:
        discount_rate = 0.05 # 5% discount
        print("Congratulations! You get a 5% discount.")
    else:
        print("You are not eligible for a discount.")

# Calculate the discount amount
discount_amount = total_purchase_amount * discount_rate

# Calculate the final amount after discount
final_amount = total_purchase_amount - discount_amount

# Print the discount amount and final payable amount
print(f"Discount Amount: Rs. {discount_amount:.2f}")
print(f"Amount to Pay after Discount: Rs. {final_amount:.2f}")

```

OUTPUT:

```

= RESTART: C:/Users/Admin/akash verma.py
Enter the total purchase amount (Rs): 10000
Congratulations! You get a 10% discount.
Discount Amount: Rs. 1000.00
Amount to Pay after Discount: Rs. 9000.00

```

PROGRAM 18:

```

# PROGRAM 18: To demonstrate the use of the for loop

number = [1, 2, 3, 4, 5]

for num in number: # use the correct variable name
    multiplied = num * 2
    print(multiplied)

```

OUTPUT:

```

= RESTART: C:/Users/Admin/akash verma.py
2
4
6
8
10

```

PROGRAM 19:

```
# Program 19: To demonstrate the use of the while loop

number = int(input("Enter a number to print its multiplication table: "))
counter = 1

print(f"\nMultiplication table of {number}:\n")

while counter <= 10:
    print(f"{number} x {counter} = {number * counter}")
    counter += 1
```

OUTPUT:

```
===== RESTART: C:/Users/Admin/akash verma.py =====
Enter a number to print its multiplication table: 17

Multiplication table of 17:

17 x 1 = 17
17 x 2 = 34
17 x 3 = 51
17 x 4 = 68
17 x 5 = 85
17 x 6 = 102
17 x 7 = 119
17 x 8 = 136
17 x 9 = 153
17 x 10 = 170
```

PROGRAM 20:

```
#program 20: to demonstrate the use of a range () with only stop value
for i in range(5):
    print(i)
```

OUTPUT:

```
= RESTART: C:/Users/Admin/akash verma.py
0
1
2
3
4
```

PROGRAM 21:

```
#program 21: to demonstrate the se of a range () with start and stop values
for i in range (20,25):
    print(i)
```

OUTPUT:

```
= RESTART: C:/Users/Admin/akashai.py
20
21
22
23
24
```

PROGRAM 21:

```
#program 22: to demonstrate the use of range () with a start,stop and step values
for i in range(25,5,-5):
    print(i)
```

OUTPUT:

```
= RESTART: C:/Users/Admin/akashai.py
25
20
15
10
```

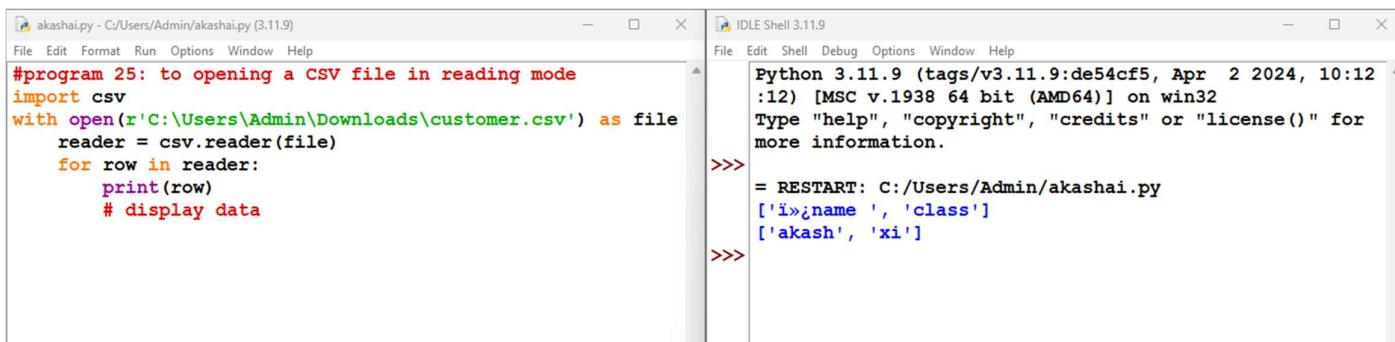
THE PROGRAM 24:

```
#program 24: to demonstrate the use of the continue statement
for i in range(1, 6):
    if i == 3 :
        continue #skip the rest of the loop body for i == 3
    print ("Iteration:", i)
```

OUTPUT:

```
===== RESTART: C:/Users/Admin/akashai.py
Iteration: 1
Iteration: 2
Iteration: 4
Iteration: 5
```

PROGRAM 25:



The image shows two windows side-by-side. The left window is titled 'akashai.py - C:/Users/Admin/akashai.py (3.11.9)' and contains Python code for reading a CSV file. The right window is titled 'IDLE Shell 3.11.9' and shows the Python interpreter shell with its version information and a few commands.

```
akashai.py - C:/Users/Admin/akashai.py (3.11.9)
File Edit Format Run Options Window Help
#program 25: to opening a CSV file in reading mode
import csv
with open(r'C:\Users\Admin\Downloads\customer.csv') as file
    reader = csv.reader(file)
    for row in reader:
        print(row)
        # display data

Python 3.11.9 (tags/v3.11.9:de54cf5, Apr  2 2024, 10:12 :12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART: C:/Users/Admin/akashai.py
['i','name ','class']
['akash', 'xi']

>>>
```

PROGRAM 27:

```
import csv # Fixed: should be lowercase 'csv'

# Open the CSV file in write mode
with open(r"C:\Users\Admin\Downloads\customer.csv", mode='w', newline='') as file:
    writer = csv.writer(file)

    # Write the header row
    writer.writerow(["customer_ID", "first_name", "last_name", "city"])

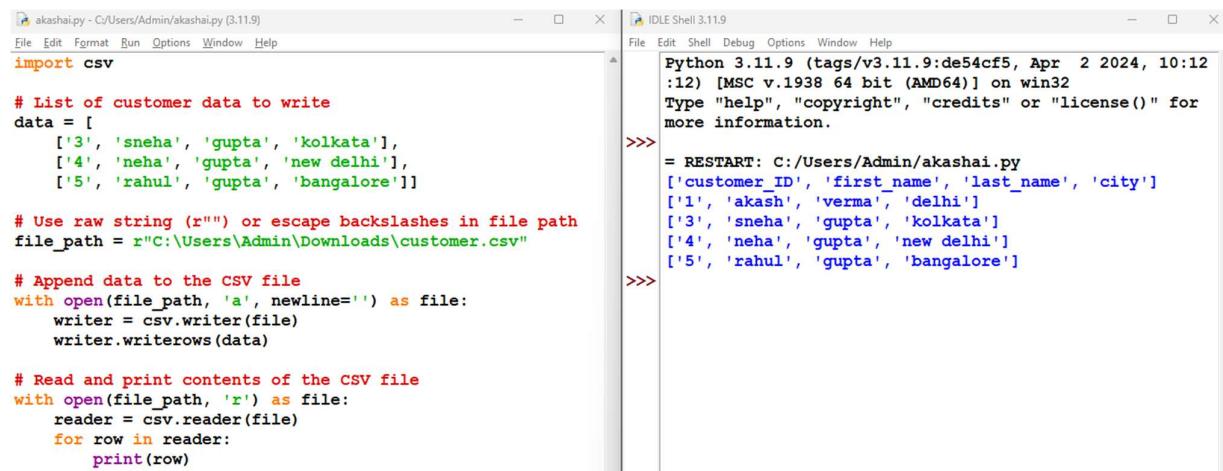
    while True: # Fixed: Python uses capital 'T' in True
        C_ID = input("Enter customer ID (type 'exit' to stop): ")
        if C_ID.lower() == 'exit': # Fixed: Comparison operator '==' instead of '='
            break
        fname = input("Enter first name: ")
        lname = input("Enter last name: ")
        city_name = input("Enter city: ")

        # Write the customer details to the CSV file
        writer.writerow([C_ID, fname, lname, city_name]) # Fixed variable names
```

OUTPUT:

```
= RESTART: C:/Users/Admin/akashai.py
Enter customer ID (type 'exit' to stop): 1
Enter first name: akash
|Enter last name: verma
Enter city: delhi
Enter customer ID (type 'exit' to stop): exit
```

PROGRAM 28:



The image shows two windows side-by-side. The left window is a code editor titled 'akashai.py - C:/Users/Admin/akashai.py (3.11.9)'. It contains Python code for reading and writing CSV files. The right window is an 'IDLE Shell 3.11.9' window showing the execution of the same code. The output in the shell shows the program reading a list of customer data from memory and then writing it back to the CSV file.

```
import csv

# List of customer data to write
data = [
    ['3', 'sneha', 'gupta', 'kolkata'],
    ['4', 'neha', 'gupta', 'new delhi'],
    ['5', 'rahul', 'gupta', 'bangalore']

# Use raw string (r "") or escape backslashes in file path
file_path = r"C:\Users\Admin\Downloads\customer.csv"

# Append data to the CSV file
with open(file_path, 'a', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(data)

# Read and print contents of the CSV file
with open(file_path, 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

```
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr  2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: C:/Users/Admin/akashai.py
['customer_ID', 'first_name', 'last_name', 'city']
['1', 'akash', 'verma', 'delhi']
['3', 'sneha', 'gupta', 'kolkata']
['4', 'neha', 'gupta', 'new delhi']
['5', 'rahul', 'gupta', 'bangalore']
```

PROGRAM 31

The screenshot shows two windows from the IDLE Python environment. The top window is titled "PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)" and contains the following Python code:

```
import numpy as np
arr_nd=np.array([[[1,2,3],[4,5,6],[7,8,9]],
                 [[10,11,12],[13,14,15],[16,17,18]],
                 [[19,20,21],[22,23,24],[25,26,27]]])
print(arr_nd)
```

The bottom window is titled "IDLE Shell 3.13.5" and shows the execution of the script. It displays the Python version, copyright information, and the command prompt ">>>". The output shows the printed 3D list:

```
>>> ===== RESTART: C:/Users/User/PROJECT ai.py =====
[[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]]

 [[10 11 12]
 [13 14 15]
 [16 17 18]]

 [[19 20 21]
 [22 23 24]
 [25 26 27]]]
```

PROGRAM 32

PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)

File Edit Format Run Options Window Help

```
import numpy as np
arr1=np.array([1,2,3,4,5])
print(arr1)
arr2=np.array([6,7,8,9,10])
print(arr2)
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 28 2023, 16:48:03) [MSC v.1934 64 bit (AMD64)]
Enter "help" below or click "Help" above
>>> ===== RESTART: C:\Users\User\PROJECT\ai.py =====
[1 2 3 4 5]
[6 7 8 9 10]
```

PROGRAM 33

PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)

File Edit Format Run Options Window Help

```
import numpy as np  
arr_zeros=np.zeros(5)  
print(arr_zeros)
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.5:6cb20a2, Ju
Enter "help" below or click "Help" above f

>>>

===== RESTART: C:

[0. 0. 0. 0. 0.]

>>> |

PROGRAM 34

PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)

File Edit Format Run Options Window Help

```
import numpy as np  
arr_ones=np.ones(5)  
print(arr_ones)
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13
Enter "help" below or click

>>>

=====

[1. 1. 1. 1. 1.]

>>>

PROGRAM 35

PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)

File Edit Format Run Options Window Help

```
import numpy as np  
arr_range=np.arange(0,10,2)  
print(arr_range)
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.
Enter "help" below or click

>>>

```
=====
```

[0 2 4 6 8]

PROGRAM 36

PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)

File Edit Format Run Options Window Help

```
import numpy as np
arr_empty=np.empty((3,3))
print(arr_empty)
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.5:6cb2
Enter "help" below or click "Help".

>>>

```
===== REST
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
```

>>> |

PROGRAM 37

```
PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)
File Edit Format Run Options Window Help
import numpy as np
rows = int(input("Enter the number of rows: "))
cols = int(input("Enter the number of columns: "))
array = np.empty((rows, cols))
for i in range(rows):
    for j in range(cols):
        value = float(input(f"Enter the value for element ({i},{j}): "))
        array[i, j] = value
print("Array:")
print(array)
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11
Enter "help" below or click "Help" above for more information
>>> ===== RESTART: C:/User
Enter the number of rows: 2
Enter the number of columns: 3
Enter the value for element (0,0): 23
Enter the value for element (0,1): 24
Enter the value for element (0,2): 6
Enter the value for element (1,0): 25
Enter the value for element (1,1): 67
Enter the value for element (1,2): 24
Array:
[[23. 24. 6.]
 [25. 67. 24.]]
```

PROGRAM 38

```
PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)
File Edit Format Run Options Window Help
import numpy as np
rows = int(input("Enter the number of rows: "))
cols=int(input("Enter the number of columns: "))
array=np.empty((rows, cols))
for i in range(rows):
    for j in range (cols):
        value = float(input(f"Enter the value for element ({i},{j}):"))
        array[i, j] = value
print("Array:")
print (array)
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:1
Enter "help" below or click "Help" above for more inform
>>>
=====
RESTART: C:/Users/User/PR
Enter the number of rows: 2
Enter the number of columns: 3
Enter the value for element (0,0):36
Enter the value for element (0,1):67
Enter the value for element (0,2):51
Enter the value for element (1,0):54
Enter the value for element (1,1):95
Enter the value for element (1,2):52
Array:
[[36. 67. 51.]
 [54. 95. 52.]]
```

PROGRAM 39

PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)

File Edit Format Run Options Window Help

```
import pandas as pd
data=[1,2,3,4,5]
series=pd.Series(data)
print(series)
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.5:6ct

Enter "help" below or click "Help"

>>>

===== RES

0 1
1 2
2 3
3 4
4 5

dtype: int64

PROGRAM 40

```
PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)
File Edit Format Run Options Window Help
import pandas as pd
data=[[1,"Mayur",25],
      [2,"Esha",30],
      [3,"Vedant",28]]
df=pd.DataFrame(data,columns=["Name","Age","City"])
print(df)
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:
Enter "help" below or click "†
>>>
=====
      Name  Age  City
 0    1   Mayur   25
 1    2    Esha   30
 2    3  Vedant   28
```

PROGRAM 41

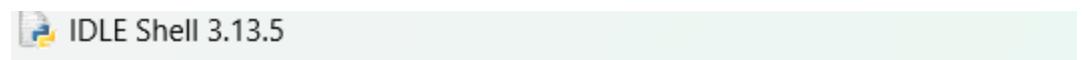
```
PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)
File Edit Format Run Options Window Help
import pandas as pd
data={"Name":["Yash","Rohan","Neha"],
      "Age":[25,30,28],
      "City":["Delhi","Mumbai","Bengaluru"]}
df=pd.DataFrame(data)
print(df)
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jul 14 2022, 15:01:33) [MSC v.1932 64 bit (AMD64)]
Enter "help" below or click "Help" above for more information.
>>> ===== RESTART: C:/Users/User/PROJECT ai.py =
   Name  Age    City
0  Yash   25  Delhi
1  Rohan  30  Mumbai
2  Neha   28 Bengaluru
```

PROGRAM 42



```
File Edit Format Run Options Window Help
import pandas as pd
import numpy as np
data=np.array([[1,"Mayur",25],
              [2,"Esha",30],
              [3,"Vedant",28]])
df=pd.DataFrame(data,columns=["ID","Name","Age"])
print(df)
```



```
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20d, Jul 10 2023, 16:08:10) [MSC v.1932 64 bit (AMD64)]
Enter "help" below or click "Help" at
>>> ===== RESTA
      ID  Name Age
0    1  Mayur  25
1    2   Esha  30
2    3  Vedant  28
...
```

PROGRAM 43

PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)

File Edit Format Run Options Window Help

```
import pandas as pd
data = [{ID: 1, Name: 'Arti ', Age: 25},
         {ID: 2, Name: 'Trinabh', Age: 30},
         {ID: 3, Name: 'Surbhi', Age: 28}]
df = pd.DataFrame (data)
print(df)
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.5:6cb20, Mar 11 2023, 15:40:29) [MSC v.1934 64 bit (AMD64)] on Windows

Enter "help" below or click "Help" ↴

>>>

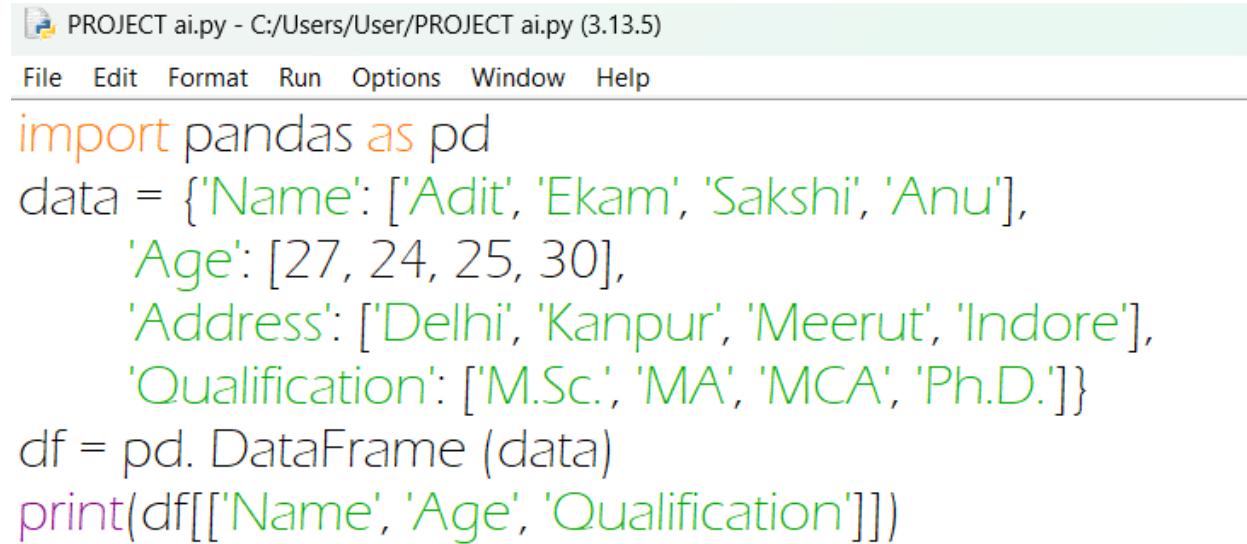
```
=====
      ID  Name  Age
0    1   Arti   25
1    2  Trinabh  30
2    3   Surbhi  28
```

PROGRAM 44

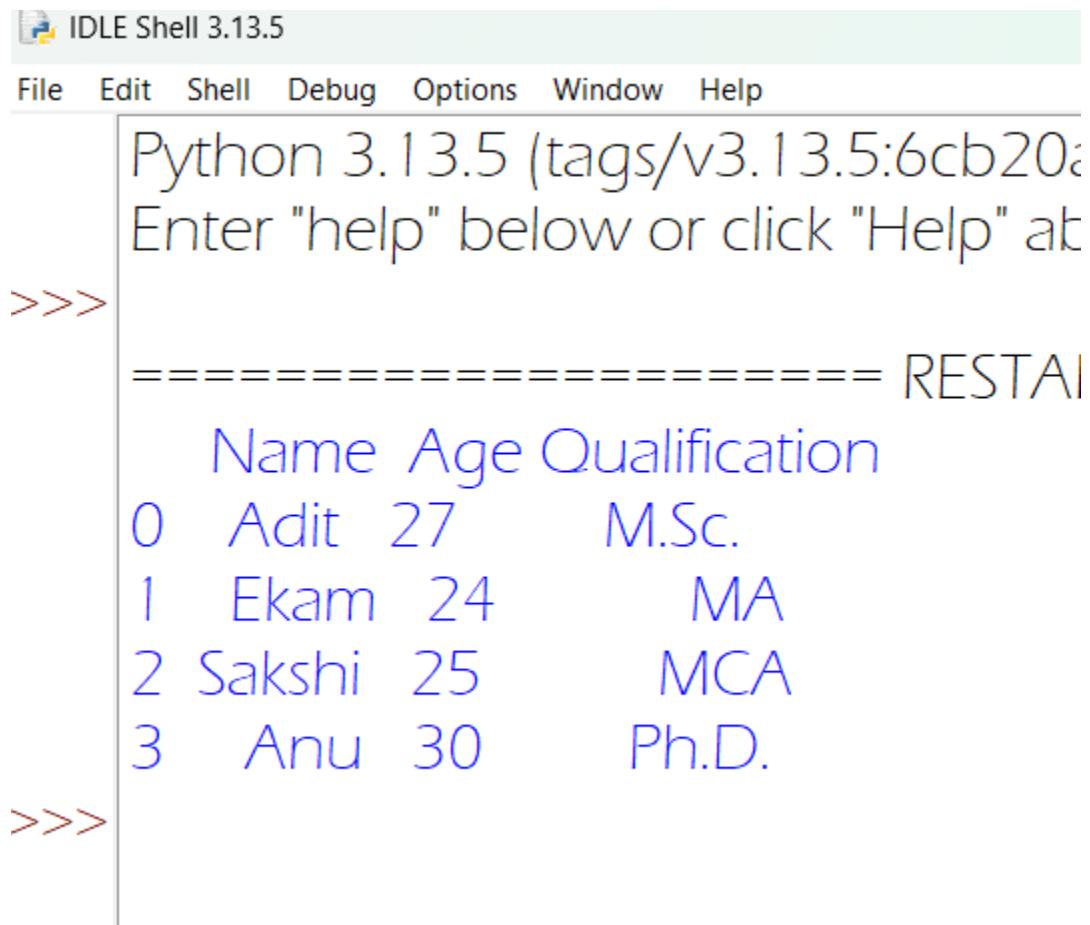
```
PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)
File Edit Format Run Options Window Help
import pandas as pd
data = {'Name': ['Adit', 'Ekam', 'Sakshi', 'Anu'],
        'Age': [27, 24, 25, 30],
        'Address': ['Delhi', 'Kanpur', 'Meerut', 'Indore'],
        'Qualification': ['M.Sc.', 'MA', 'MCA', 'Ph.D.']}
df = pd.DataFrame (data)
print(df)
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20a2, Ju
Enter "help" below or click "Help" above f
>>>
=====
RESTART: C:
      Name  Age Address Qualification
0    Adit   27    Delhi      M.Sc.
1    Ekam   24   Kanpur       MA
2   Sakshi   25   Meerut      MCA
3     Anu   30   Indore      Ph.D.
```

PROGRAM 45



```
PROJECT ai.py - C:/Users/User/PROJECT ai.py (3.13.5)
File Edit Format Run Options Window Help
import pandas as pd
data = {'Name': ['Adit', 'Ekam', 'Sakshi', 'Anu'],
        'Age': [27, 24, 25, 30],
        'Address': ['Delhi', 'Kanpur', 'Meerut', 'Indore'],
        'Qualification': ['M.Sc.', 'MA', 'MCA', 'Ph.D.']}
df = pd.DataFrame (data)
print(df[['Name', 'Age', 'Qualification']])
```



```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20d, Jul 10 2023, 16:08:00) [MSC v.1932 64 bit (AMD64)]
Enter "help" below or click "Help" at the top
>>> ===== RESTART: C:/Users/User/PROJECT ai.py =====
      Name  Age Qualification
0    Adit   27       M.Sc.
1    Ekam   24         MA
2  Sakshi   25         MCA
3     Anu   30       Ph.D.
>>>
```

```
d python.py - C:/Users/User/d python.py (3.13.5)
File Edit Format Run Options Window Help
import pandas as pd
data={"Name":["Adit","Ekam","Sakshi","Anu"],
      "Age":[27,24,25,30],
      "Address":["Delhi","Kanpur","Meerut","Indore"],
      "Qualification":["M.Sc.","MA","MCA","Ph.D."]}
df=pd.DataFrame(data)
selected_row=df.iloc[2]
print(selected_row)
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20a2, J
Enter "help" below or click "Help" above
>>>
=====
RESTART:
Name      Sakshi
Age       25
Address   Meerut
Qualification   MCA
Name: 2, dtype: object
>>>
```

d python.py - C:/Users/User/d python.py (3.13.5)

File Edit Format Run Options Window Help

```
import pandas as pd
data={"Name":["Adit","Ekam","Sakshi","Anu"],
      "Age":[27,24,25,30],
      "Address":["Delhi","Kanpur","Meerut","Indore"],
      "Qualification":["M.Sc.","MA","MCA","Ph.D."]}
df=pd.DataFrame(data)
selected_row=df.iloc[2]
selected_row_dict=selected_row.to_dict()
selected_row_df=pd.DataFrame([selected_row_dict])
print(selected_row_df)
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

```
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11
Enter "help" below or click "Help" above for n
>>>
=====
RESTART: C:/Us
      Name  Age Address Qualification
0  Sakshi   25    Meerut        MCA
>>>
```

```
d python.py - C:/Users/User/d python.py (3.13.5)
File Edit Format Run Options Window Help
import pandas as pd
data={"Name":["Adit","Ekam","Sakshi","Anu"],
      "Age":[27,24,25,30],
      "Address":["Delhi","Kanpur","Meerut","Indore"],
      "Qualification":["M.Sc.","MA","MCA","Ph.D."]}
df=pd.DataFrame(data)
element=df.iloc[1,2]
print("Element at (1,2):",element)
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20a2, J
Enter "help" below or click "Help" above
>>>
=====
RESTART:
Element at (1,2): Kanpur
>>> |
```

```
d python.py - C:/Users/User/d python.py (3.13.5)
File Edit Format Run Options Window Help
import pandas as pd
data={"Name":["Adit","Ekam","Sakshi","Anu"],
      "Age":[27,24,25,30],
      "Address":["Delhi","Kanpur","Meerut","Indore"],
      "Qualification":["M.Sc.","MA","MCA","Ph.D."]}
df=pd.DataFrame(data)
df['Salary']=[50000,45000,55000,60000]
print(df[["Name","Age","Address","Qualification","Salary"]])
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 1
Enter "help" below or click "Help" above for more info
>>>
=====
RESTART: C:/Users/User
      Name  Age Address Qualification  Salary
0    Adit   27   Delhi      M.Sc.  50000
1    Ekam   24  Kanpur        MA  45000
2   Sakshi   25  Meerut       MCA  55000
3     Anu   30  Indore      Ph.D.  60000
>>>
```

```
d python.py - C:/Users/User/d python.py (3.13.5)
File Edit Format Run Options Window Help
import pandas as pd
data={"Name":["Adit","Ekam","Sakshi","Anu"],
      "Age":[27,24,25,30],
      "Address":["Delhi","Kanpur","Meerut","Indore"],
      "Qualification":["M.Sc.","MA","MCA","Ph.D."]}
df=pd.DataFrame(data)
Salary=[50000,45000,55000,60000]
Index_position=2
df.insert(Index_position,"Salary",Salary)
print(df)
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025,
Enter "help" below or click "Help" above for more info
>>>
=====
RESTART: C:/Users/User/
      Name  Age  Salary Address Qualification
0    Adit   27    50000   Delhi        M.Sc.
1    Ekam   24    45000  Kanpur        MA
2   Sakshi   25    55000  Meerut        MCA
3     Anu   30    60000  Indore        Ph.D.
>>> |
```

```
d python.py - C:/Users/User/d python.py (3.13.5)
File Edit Format Run Options Window Help
import pandas as pd
data={"Name":["Adit","Ekam","Sakshi","Anu"],
      "Age":[27,24,25,30],
      "Address":["Delhi","Kanpur","Meerut","Indore"],
      "Qualification":["M.Sc.","MA","MCA","Ph.D."]}
df=pd.DataFrame(data)
df["Salary"]=[50000,45000,55000,60000]
new_row={'Name': 'Himanshi', 'Age': 29, 'Address': 'Mumbai', 'Qualification': 'B.Tech', 'Salary': 52000}
df.loc [len (df)] = new_row
print("DataFrame with the new row:")
print (df)
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:00:00) [MSC v.1938 64 bit (AMD64)]
Enter "help" below or click "Help" above for more information.
>>>
=====
RESTART: C:/Users/User/d python.py
DataFrame with the new row:
   Name  Age Address Qualification  Salary
0   Adit   27    Delhi       M.Sc.  50000
1   Ekam   24   Kanpur        MA  45000
2  Sakshi   25   Meerut       MCA  55000
3    Anu   30   Indore      Ph.D.  60000
4  Himanshi   29   Mumbai     B.Tech  52000
```

```
d python.py - C:/Users/User/d python.py (3.13.5)
File Edit Format Run Options Window Help
import pandas as pd
data={"Name":["Adit","Ekam","Sakshi","Anu"],
      "Age":[27,24,25,30],
      "Address":["Delhi","Kanpur","Meerut","Indore"],
      "Qualification":["M.Sc.","MA","MCA","Ph.D."]}
df=pd.DataFrame(data)
df["Salary"]=[50000,45000,55000,60000]
new_row={'Name': 'Himanshi', 'Age': 29, 'Address': 'Mumbai', 'Qualification': 'B.Tech', 'Salary': 52000}
index_to_insert = len (df)//2
df = pd.concat([df.iloc[:index_to_insert], pd.DataFrame([new_row]), df.iloc[index_to_insert:]]).reset_index (drop=True)
print("DataFrame with the new row in the middle:")
print (df)
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025,
Enter "help" below or click "Help" above for more info

>>>

===== RESTART: C:/Users/User/d python.py (3.13.5)*

DataFrame with the new row in the middle:

	Name	Age	Address	Qualification	Salary
0	Adit	27	Delhi	M.Sc.	50000
1	Ekam	24	Kanpur	MA	45000
2	Himanshi	29	Mumbai	B.Tech	52000
3	Sakshi	25	Meerut	MCA	55000
4	Anu	30	Indore	Ph.D.	60000

d python.py - C:/Users/User/d python.py (3.13.5)

File Edit Format Run Options Window Help

```
import pandas as pd
data={"Name":["Adit","Ekam","Sakshi","Anu"],
      "Age":[27,24,25,30],
      "Address":["Delhi","Kanpur","Meerut","Indore"],
      "Qualification":["M.Sc.","MA","MCA","Ph.D."]}
df=pd.DataFrame(data)
df["Salary"]=[50000,45000,55000,60000]
df.drop(index=1, inplace=True)
print("\nDataFrame after deleting a row:")
print (df)
df.drop(columns='Address', inplace=True)
print("DataFrame after deleting a column:")
print (df)
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:12:47) [MSC v.1938 64 bit (AMD64)] on win32
Type "help" for help.

>>>

===== RESTART: C:/Users/User/

DataFrame after deleting a row:

	Name	Age	Address	Qualification	Salary
0	Adit	27	Delhi	M.Sc.	50000
2	Sakshi	25	Meerut	MCA	55000
3	Anu	30	Indore	Ph.D.	60000

DataFrame after deleting a column:

	Name	Age	Qualification	Salary
0	Adit	27	M.Sc.	50000
2	Sakshi	25	MCA	55000
3	Anu	30	Ph.D.	60000

>>>

d python.py - C:/Users/User/d python.py (3.13.5)

File Edit Format Run Options Window Help

```
import pandas as pd
import numpy as np
data={"Name":["Adit","Ekam","Sakshi","Anu"],
      "Age":[27,np.nan,25,30],
      "Address":["Delhi","Kanpur","Meerut","Indore"],
      "Qualification":["M.Sc.","MA","MCA","Ph.D."]}
df=pd.DataFrame(data)
print("Original DataFrame:")
print(df)
print("\nMissing values in each column: ")
print(df.isnull().sum())
print(df.isnull().sum().sum())
df_dropped=df.dropna()
print("\nDataFrame after dropping rows with NaN values:")
print(df_dropped)
df_filled=df.fillna({'Age': round(df ['Age'].mean()), 'Address': 'Chennai'})
print("\nDataFrame after filling NaN values:")
print(df_filled)
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
>>>
=====
RESTART: C:/Users/User/d python.py =====
Original DataFrame:
   Name  Age Address Qualification
0  Adit  27.0    Delhi      M.Sc.
1  Ekam   NaN  Kanpur        MA
2 Sakshi  25.0  Meerut       MCA
3   Anu  30.0  Indore     Ph.D.

Missing values in each column:
Name      0
Age       1
Address   0
Qualification  0
dtype: int64
1

DataFrame after dropping rows with NaN values:
   Name  Age Address Qualification
0  Adit  27.0    Delhi      M.Sc.
2 Sakshi  25.0  Meerut       MCA
3   Anu  30.0  Indore     Ph.D.

DataFrame after filling NaN values:
   Name  Age Address Qualification
0  Adit  27.0    Delhi      M.Sc.
1  Ekam  27.0  Kanpur        MA
2 Sakshi  25.0  Meerut       MCA
3   Anu  30.0  Indore     Ph.D.

>>>
```

d python.py - C:/Users/User/d python.py (3.13.5)

File Edit Format Run Options Window Help

```
import pandas as pd
data = {
    'Product': ['Laptop', 'Tablet', 'Smartphone', 'Monitor'],
    'Price': [1000, 500, 800, 300],
    'Stock': [50, 150, 200, 100],
    'Rating': [4.5, 4.0, 4.7, 4.3]
}
df = pd.DataFrame (data)
print(df)
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11
Enter "help" below or click "Help" above for n

>>>

===== RESTART: C:/Use

	Product	Price	Stock	Rating
0	Laptop	1000	50	4.5
1	Tablet	500	150	4.0
2	Smartphone	800	200	4.7
3	Monitor	300	100	4.3

d python.py - C:/Users/User/d python.py (3.13.5)

File Edit Format Run Options Window Help

```
import pandas as pd
df = pd.read_csv("Customer.csv", sep=',', header=0)
print (df)
```

Customer ID,First Name,Last Name,City

1,Akash,Patel,Mumbai
2,Priya,Sharma,Delhi
3,Aarav,Singh,Jaipur
4,Neha,Trivedi,Ahmedabad
5,Rahul,Mehta,Bangalore
6,Aishwarya,Iyer,Chennai
7,Rajesh,Reddy,Hyderabad
8,Sneha,Gupta,Kolkata
9,Sanjay,Kumar,Lucknow
10,Deepika,Menon,Pune

d python.py - C:/Users/User/d python.py (3.13.5)

File Edit Format Run Options Window Help

```
import pandas as pd
data = {'Name': ['Adit', 'Ekam', 'Sakshi', 'Anu'],
        'Age': [27, 24, 25, 30],
        'Address': ['Delhi', 'Kanpur', 'Meerut', 'Indore'],
        'Qualification': ['M.Sc.', 'MA', 'MCA', 'Ph.D.']}
df=pd.DataFrame (data)
df.to_csv('employee_data.csv', sep=',', index=False)
```

d python.py - C:/Users/User/d python.py (3.13.5)

File Edit Format Run Options Window Help

```
from sklearn.datasets import load_iris
iris = load_iris()
print(iris.data[:10])
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.5:6cb20...
Enter "help" below or click "Help" ↴

>>>

```
===== RESTA
[[5.1 3.5 1.4 0.2]
 [4.9 3. 1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5. 3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5. 3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]]
```

d python.py - C:/Users/User/d python.py (3.13.5)

File Edit Format Run Options Window Help

```
from sklearn.datasets import load_iris
iris=load_iris()
X=iris.data
y=iris.target
```



d python.py - C:/Users/User/d python.py (3.13.5)

File Edit Format Run Options Window Help

```
from sklearn.datasets import load_iris
iris=load_iris()
x = iris.data
y = iris.target
print("Features (X):")
print ( x [:10])
print("\nTarget (y):")
print ( y [:10])
```



IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.5:6cb20a2, Ju
Enter "help" below or click "Help" above

>>>

===== RESTART: C:

Features (X):

[[5.1 3.5 1.4 0.2]
[4.9 3. 1.4 0.2]
[4.7 3.2 1.3 0.2]
[4.6 3.1 1.5 0.2]
[5. 3.6 1.4 0.2]
[5.4 3.9 1.7 0.4]
[4.6 3.4 1.4 0.3]
[5. 3.4 1.5 0.2]
[4.4 2.9 1.4 0.2]
[4.9 3.1 1.5 0.1]]

Target (y):

[0 0 0 0 0 0 0 0]

```
d python.py - C:/Users/User/d python.py (3.13.5)
File Edit Format Run Options Window Help
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
iris = load_iris()
x = iris.data
y = iris.target
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=1)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
print("Training set Features:", x_train.shape, " Labels:", y_train.shape)
print("Testing set Features:", x_test.shape, " Labels:", y_test.shape)
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025,
Enter "help" below or click "Help" above for more i
>>>
=====
RESTART: C:/Users/Us
Training set Features: (120, 4) Labels: (120,)
Testing set Features: (30, 4) Labels: (30,)
>>>
```

```
d python.py - C:/Users/User/d python.py (3.13.5)
File Edit Format Run Options Window Help
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
iris=load_iris()
x=iris.data
y=iris.target
x_train, x_test, y_train, y_test=train_test_split(x, y,test_size=0.2, random_state=1)
knn=KNeighborsClassifier(n_neighbors=3)
knn.fit(x_train,y_train)
y_pred=knn.predict(x_test)
```

d python.py - C:/Users/User/d python.py (3.13.5)

File Edit Format Run Options Window Help

```
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
iris=load_iris()
X=iris.data
y=iris.target
X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.2, random_state=1)
X_train, X_test, y_train,y_test=train_test_split(X, y, test_size=0.2, random_state=1)
knn=KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
y_pred=knn.predict (X_test)
accuracy=metrics.accuracy_score(y_test, y_pred)
precision=metrics.precision_score (y_test, y_pred, average="weighted")
print("Accuracy:", accuracy)
print("Precision:", precision)
```

IDLE Shell 3.13.5

File Edit Shell Debug Options Window Help

Python 3.13.5 (tags/v3.13.5:6cb...

Enter "help" below or click "Help"

>>>

===== RES

Accuracy: 1.0

Precision: 1.0

>>> |

```
d python.py - C:/Users/User/d python.py (3.13.5)
File Edit Format Run Options Window Help
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
iris=load_iris()
X=iris.data
y=iris.target
X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.2,random_state=1)
X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.2,random_state=1)
knn=KNeighborsClassifier (n_neighbors=3)
knn.fit(X_train, y_train)
y_pred=knn.predict(X_test)
accuracy=metrics.accuracy_score(y_test, y_pred)
sample=[[3, 5, 4, 2], [2, 2, 5, 4]]
prediction=knn.predict(sample)
prediction_species=[iris.target_names[p] for p in prediction]
print("Accuracy:", accuracy)
print("Predictions:", prediction_species)
```

```
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:00:00) [MSC v.1938 64 bit (AMD64)]
Enter "help" below or click "Help" above for more information.
>>> ===== RESTART: C:/Users/User/d python.py =
Accuracy: 1.0
Predictions: [np.str_('versicolor'), np.str_('virginica')]
>>>
```

Program 1 :-

The screenshot shows two windows from Python IDLE. The left window is a code editor with the file 'akash123.py' containing Python code to calculate the mean height of students. The right window is the 'IDLE Shell 3.11.9' window showing the execution results.

```
akash123.py - C:/Users/Admin/akash123.py (3.11.9)
File Edit Format Run Options Window Help
import statistics
height = [145,151, 152, 149, 147, 152, 151,149,
152, 151, 147, 148, 155, 147, 152, 151,
149,145, 147, 152, 146, 148, 150, 152, 151]
print ("Mean height of students", statistics.mean (height))

IDLE Shell 3.11.9
File Edit Shell Debug Options Window Help
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12 :12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/akash123.py
Mean height of students 149.56
```

Program 2 :-

The screenshot shows two windows from Python IDLE. The left window is a code editor with the file 'akash123.py' containing Python code to calculate the median height of students. The right window is the 'IDLE Shell 3.11.9' window showing the execution results.

```
akash123.py - C:/Users/Admin/akash123.py (3.11.9)
File Edit Format Run Options Window Help
import statistics
height = [145,151, 152, 149, 147, 152, 151,149,
152, 151, 147, 148, 155, 147, 152, 151,
149,145, 147, 152, 146, 148, 150, 152, 151]
print ("Median of height of students", statistics.median (height))

IDLE Shell 3.11.9
File Edit Shell Debug Options Window Help
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12 :12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/akash123.py
Median of height of students 150
>>>
```

Program 3 :-

The screenshot shows two windows from Python IDLE. The left window is a code editor with the file 'akash123.py' containing Python code to calculate the median height of students. The right window is the 'IDLE Shell 3.11.9' window showing the execution results.

```
akash123.py - C:/Users/Admin/akash123.py (3.11.9)
File Edit Format Run Options Window Help
import statistics
height = [145,151, 152, 149, 147, 152, 151,149,
152, 151, 147, 148, 155, 147, 152, 151,
149,145, 147, 152, 146, 148, 150, 152, 151]
print ("Median of height of students", statistics.median (height))

IDLE Shell 3.11.9
File Edit Shell Debug Options Window Help
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12 :12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/akash123.py
Median of height of students 150
>>>
```

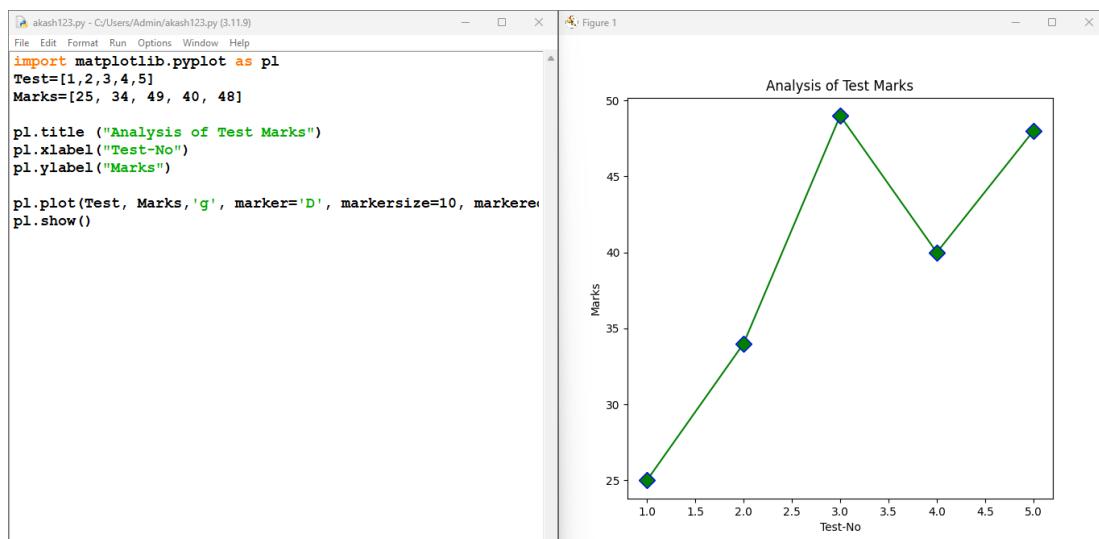
Program 4 :-

The screenshot shows two windows from Python IDLE. The left window is a code editor with the file 'akash123.py' containing Python code to calculate the mode height of students. The right window is the 'IDLE Shell 3.11.9' window showing the execution results.

```
akash123.py - C:/Users/Admin/akash123.py (3.11.9)
File Edit Format Run Options Window Help
import statistics
height = [145,151, 152, 149, 147, 152, 151, 149,
152, 151, 147, 148, 155, 147, 152, 151,
149,145, 147, 152, 146, 148, 150, 152, 151]
print ("Mode of height of students", statistics.mode (height))

IDLE Shell 3.11.9
File Edit Shell Debug Options Window Help
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12 :12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/akash123.py
Mode of height of students 152
```

Program 5 :-

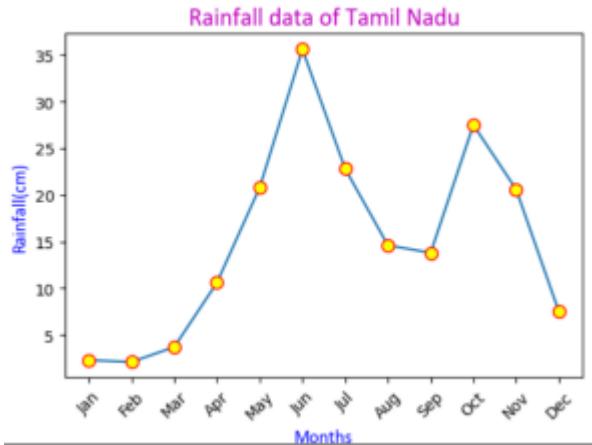


Program 6 :-

```

import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("rainfall.csv")
x=df['Months']
y=df['Rainfall(cm)']
plt.figure(figsize=(6,4))
plt.plot(x,y,marker='o', markersize=8, markeredgecolor='red', markerfacecolor='yellow')
plt.xticks(rotation = 45)
plt.xlabel("Months",fontname='Calibri',color='b',fontsize=12)
plt.ylabel("Rainfall(cm)",fontname='Calibri',color='b',fontsize=12)
plt.title("Rainfall data of Tamil Nadu",fontname='Calibri',color='m',fontsize=16)
plt.show()

```

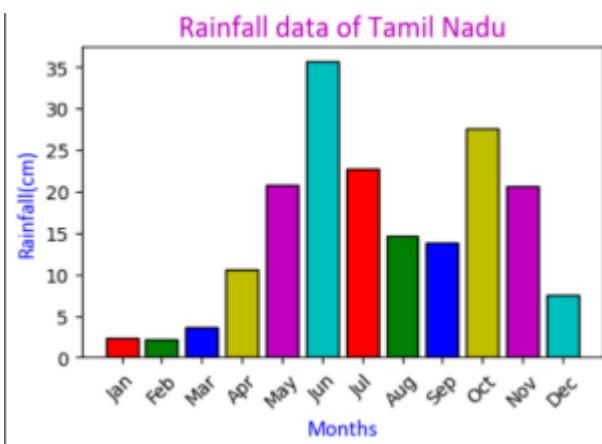


Program 7:-

```

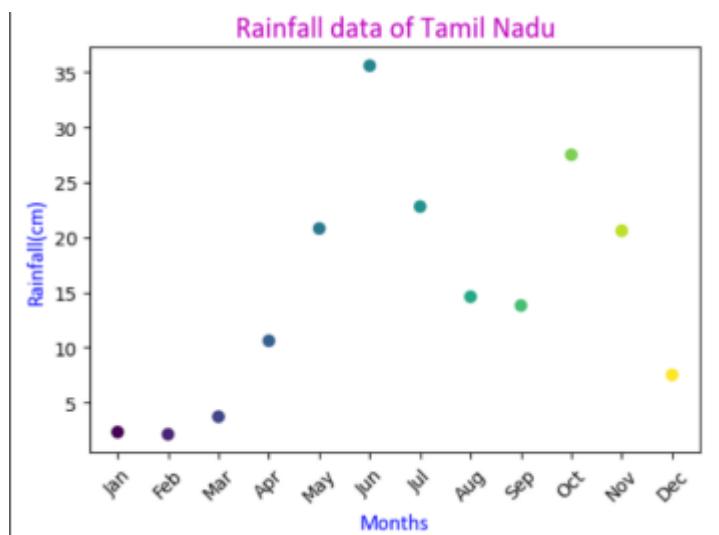
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("rainfall.csv")
x=df['Months']
y=df['Rainfall(cm)']
plt.figure(figsize=(6,4))
c=['r','g','b','y','m','c']
plt.bar(x,y,color=c,edgecolor='k')
plt.xticks(rotation = 45)
plt.xlabel("Months",fontname='Calibri',color='b',fontsize=12)
plt.ylabel("Rainfall(cm)",fontname='Calibri',color='b',fontsize=12)
plt.title("Rainfall data of Tamil Nadu",fontname='Calibri',color='m',fontsize=16)
plt.show()

```



Program 8:-

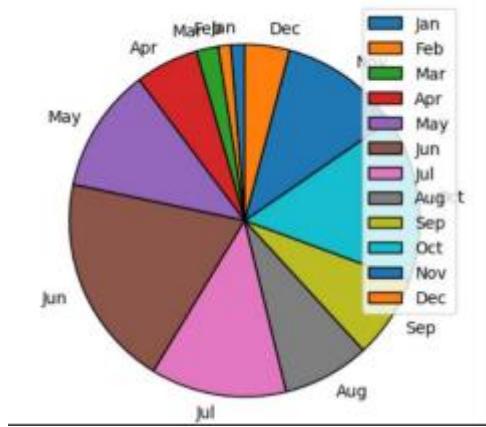
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("rainfall.csv")
x=df['Months']
y=df['Rainfall(cm)']
plt.figure(figsize=(5,3))
plt.figure(figsize=(6,4))
colors = np.array([0, 10, 20, 30, 40, 45, 50, 60, 70, 80, 90, 100])
plt.scatter(x,y,c=colors,cmap='viridis')
plt.title("Rainfall data of Tamil Nadu",fontname='Calibri',color='m',fontsize=16)
plt.xticks(rotation = 45)
plt.xlabel("Months",fontname='Calibri',color='b',fontsize=12)
plt.ylabel("Rainfall(cm)",fontname='Calibri',color='b',fontsize=12)
plt.show()
```



Program 9:-

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("rainfall.csv")
x=df['Months']
y=df['Rainfall(cm)']
wp = { 'linewidth' : 1, 'edgecolor' : "black" }
plt.pie(y,labels=x, startangle=90,wedgeprops=wp)
plt.legend(loc='upper right')
plt.title("Rainfall data of Tamil Nadu",fontname='Calibri',color='m',fontsize=16)
plt.show()
```

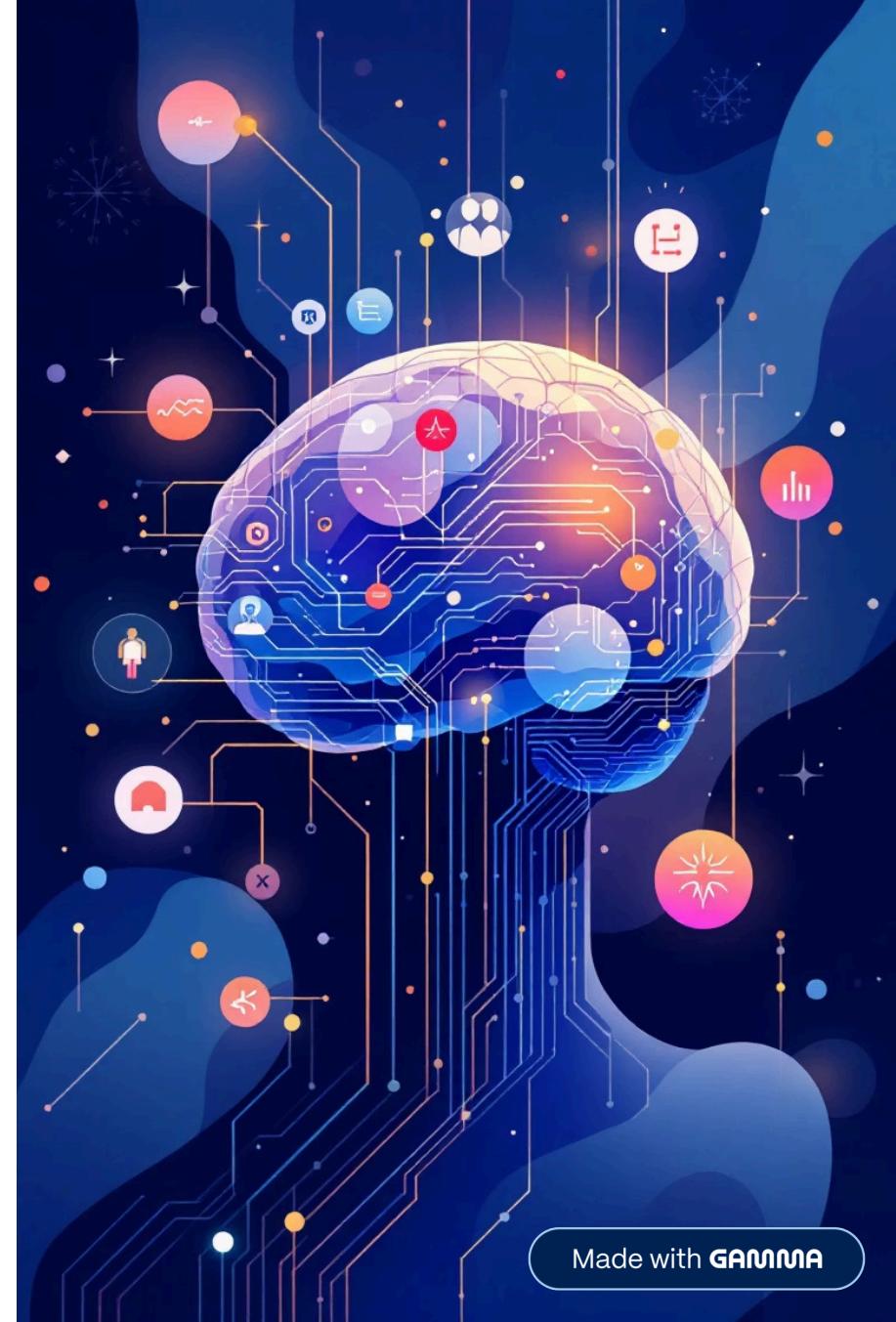
Rainfall data of Tamil Nadu



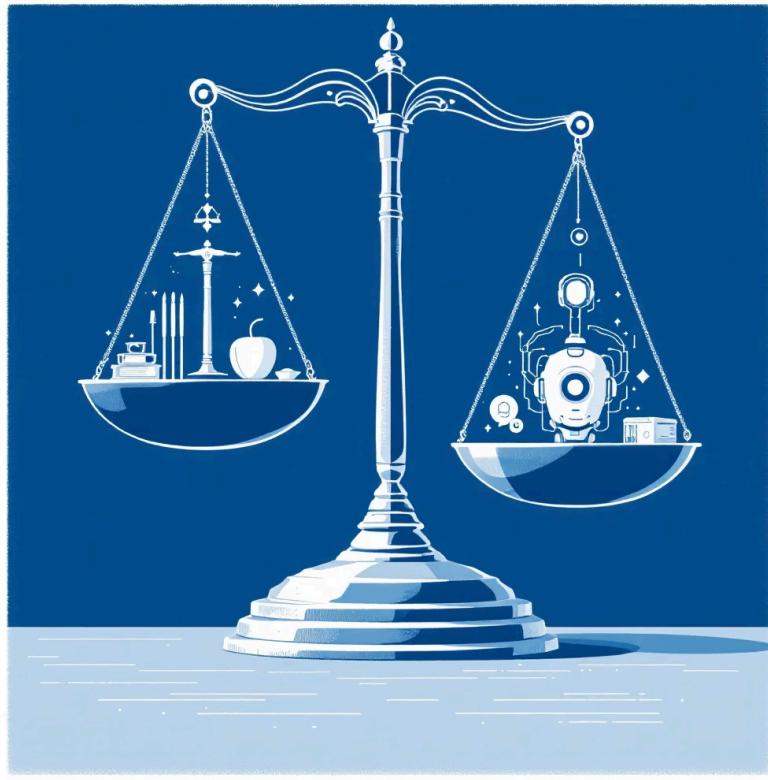
AI Ethics and Values

Ensuring Responsible Use of Artificial Intelligence

Understanding how we can build AI systems that are fair, safe, and beneficial for everyone.



What Is AI Ethics?



A Framework for Responsible AI

AI ethics refers to the moral principles and guidelines that govern how artificial intelligence systems are developed, deployed, and used. It ensures that AI benefits society while minimizing potential harms.

As AI becomes more powerful and widespread, we need clear ethical guidelines to ensure these systems respect human rights, promote fairness, and operate transparently.

Why Ethics Matter in AI

Artificial intelligence is rapidly transforming every aspect of our lives, making ethical considerations more critical than ever before.



Healthcare

AI diagnoses diseases and recommends treatments, affecting life-or-death decisions.



Finance

AI systems determine loan approvals, credit scores, and investment strategies.



Education

AI personalizes learning experiences and evaluates student performance.

Without proper ethical frameworks, AI systems could perpetuate discrimination, invade privacy, or make decisions that harm individuals and communities.

Core Ethical Principles



Fairness & Avoiding Bias

AI systems should treat all users equitably, regardless of race, gender, age, or background.



Transparency

People should understand how AI systems make decisions that affect them.



Accountability

Developers and organizations must take responsibility for AI system outcomes.



Privacy

Personal data must be protected and used only with proper consent.



Safety & Reliability

AI systems must function correctly and avoid causing harm to users.

Understanding Bias in AI

What Is Bias?

Bias occurs when an AI system consistently produces unfair outcomes that favor or disadvantage certain groups of people. This happens when training data reflects existing societal prejudices or lacks diversity.

How It Happens

- Training data contains historical discrimination
- Dataset lacks representation of certain groups
- Algorithms amplify existing patterns in data

Real-World Examples



Hiring Systems

An AI recruitment tool was found to favor male candidates because it was trained on historical hiring data that reflected gender bias in the tech industry.



Facial Recognition

Studies show that facial recognition systems have higher error rates for people with darker skin tones, leading to unfair outcomes in security and identification.



Privacy & Data Protection

What Is Personal Data?

Personal data includes any information that can identify you: name, photos, location, browsing history, health records, and even your preferences and behavior patterns.

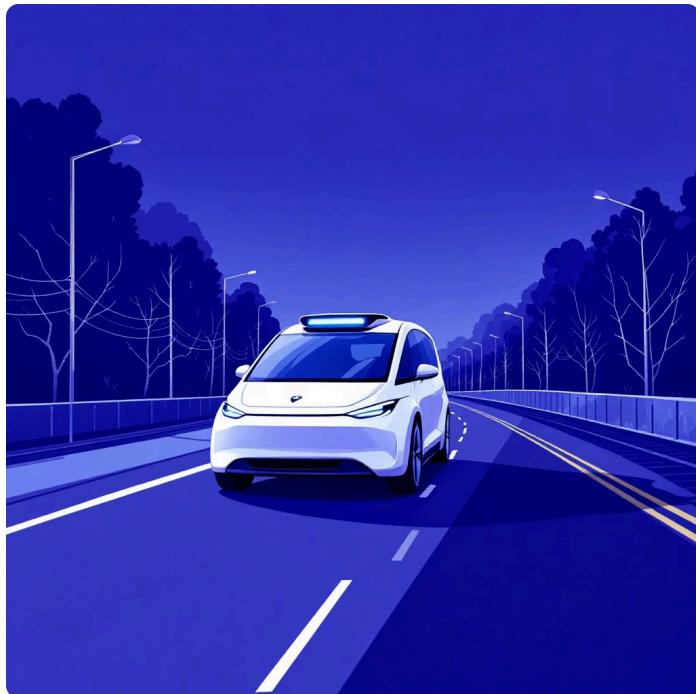
Why Privacy Matters

AI systems often require massive amounts of data to function. Without proper protection, this data can be misused for unauthorized surveillance, identity theft, or manipulation.

Key Protections

- Informed consent:** Users must know what data is collected and how it's used
- Data security:** Strong encryption and safeguards prevent breaches
- Minimal collection:** Only gather data that's truly necessary

Safety & Trustworthiness



Building Reliable AI Systems

For AI to be trusted in critical applications, it must be safe, reliable, and explainable.

- **Reliability**

The system performs consistently and accurately across different situations and conditions.

- **Explainability**

Humans can understand why the AI made specific decisions, enabling oversight and correction.

- **Robustness**

The AI handles unexpected inputs and edge cases without catastrophic failures.

Examples: Self-driving cars must reliably detect obstacles in all weather conditions. Medical AI tools must explain their diagnostic reasoning so doctors can validate recommendations.

Ethical Challenges We Face

As AI technology advances, new ethical dilemmas emerge that society must address.

Deepfakes

AI-generated fake videos and audio can spread misinformation, damage reputations, and undermine trust in media.

Misinformation

AI can rapidly generate and spread false information, making it harder to distinguish truth from fiction.

Job Displacement

Automation powered by AI may eliminate certain jobs, requiring workers to adapt and retrain.

Surveillance Risks

AI-powered surveillance systems can enable mass monitoring, threatening civil liberties and freedom.



Responsible AI Practices



Ethical Guidelines

Establish clear principles and frameworks that guide AI development from conception to deployment.



Human Oversight

Keep humans in the decision-making loop, especially for high-stakes choices affecting people's lives.



Regular Audits

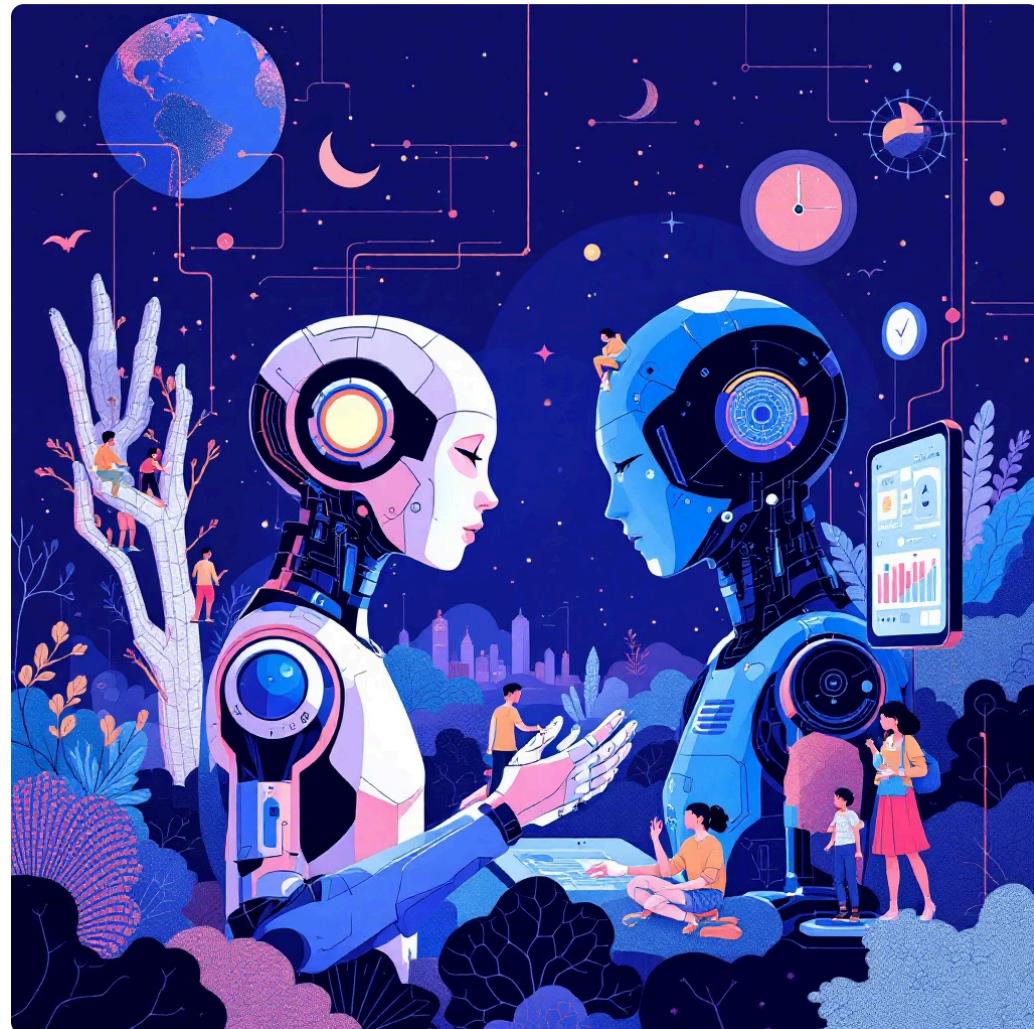
Continuously test AI systems for bias, errors, and unintended consequences, making corrections as needed.



Transparent Explanations

Provide clear explanations of how AI models work and why they make specific recommendations.

Moving Forward Responsibly



AI should augment, not replace, human judgment.

As we develop increasingly powerful AI systems, we must ensure they serve humanity's best interests. This requires ongoing dialogue between technologists, policymakers, ethicists, and citizens.

Remember: Every one of us—as students, developers, and users—has a role in shaping the ethical future of artificial intelligence.

[Learn More About AI Ethics](#)



AI in Smart Learning: Transforming Education with Artificial Intelligence

Why AI Matters in Education Today

The Challenge: Over 50% of teachers experience burnout due to heavy workloads, impacting educational quality and teacher retention.

AI offers a powerful solution by automating administrative tasks, freeing up valuable time for educators to focus on teaching.

Student Benefits: Students receive personalized learning experiences, tailored precisely to their unique pace and needs, fostering deeper engagement and improved outcomes.



What is Smart Learning with AI?



Adaptive Learning

AI tools dynamically adjust lessons to each student's individual pace and learning style, ensuring optimal comprehension.



Enhanced Feedback

Real-time performance data and feedback empower teachers to provide targeted support, intervening precisely when needed.



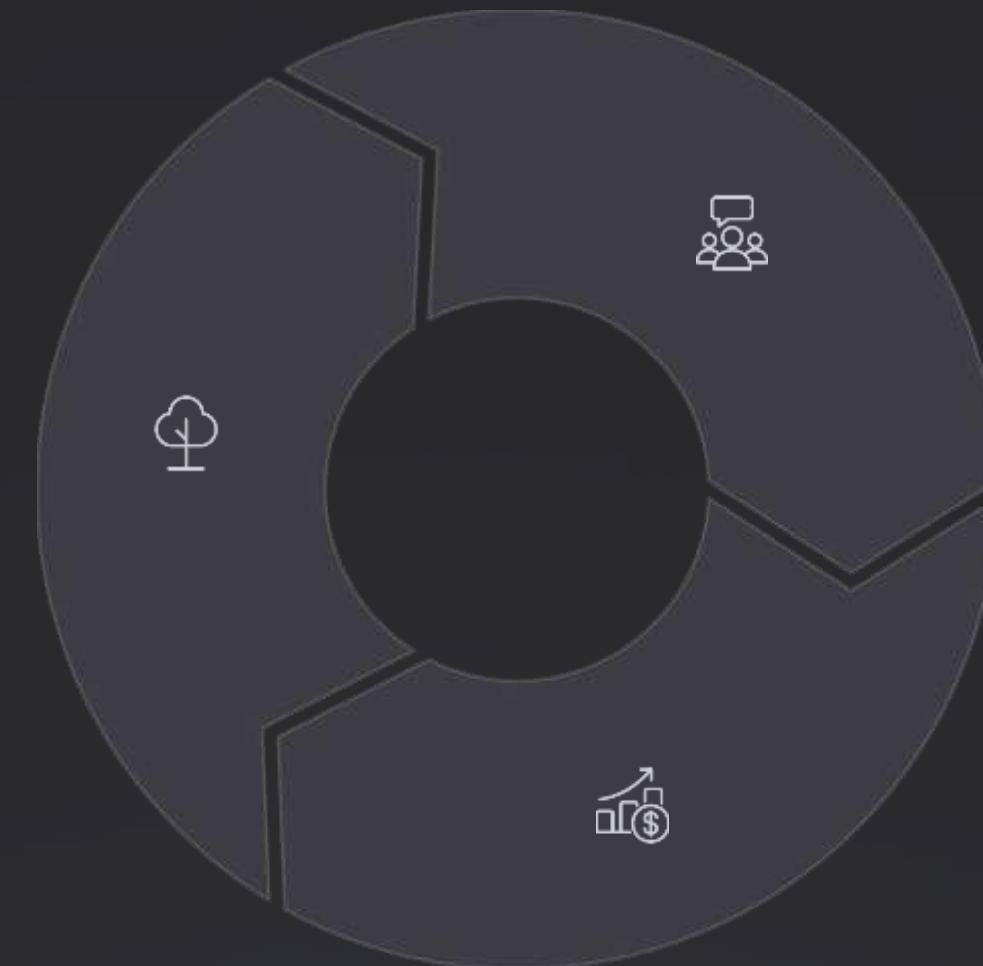
Automated Assistance

Examples include AI tutors providing instant help, automated grading systems, and smart content recommendations for students.

Key Components of an AI Smart Learning System

Personalized Paths

Machine learning algorithms create unique learning trajectories for every student.



NLP for Feedback

Natural Language Processing automates feedback and assessment, reducing manual workload.

Data Dashboards

Teachers gain instant, visual insights into student progress, enabling proactive intervention.

Case Study: TeachSmart AI Tutoring System

Teacher Empowerment

TeachSmart reduces educator stress by automating time-consuming tasks like attendance tracking, grading assignments, and generating progress reports.

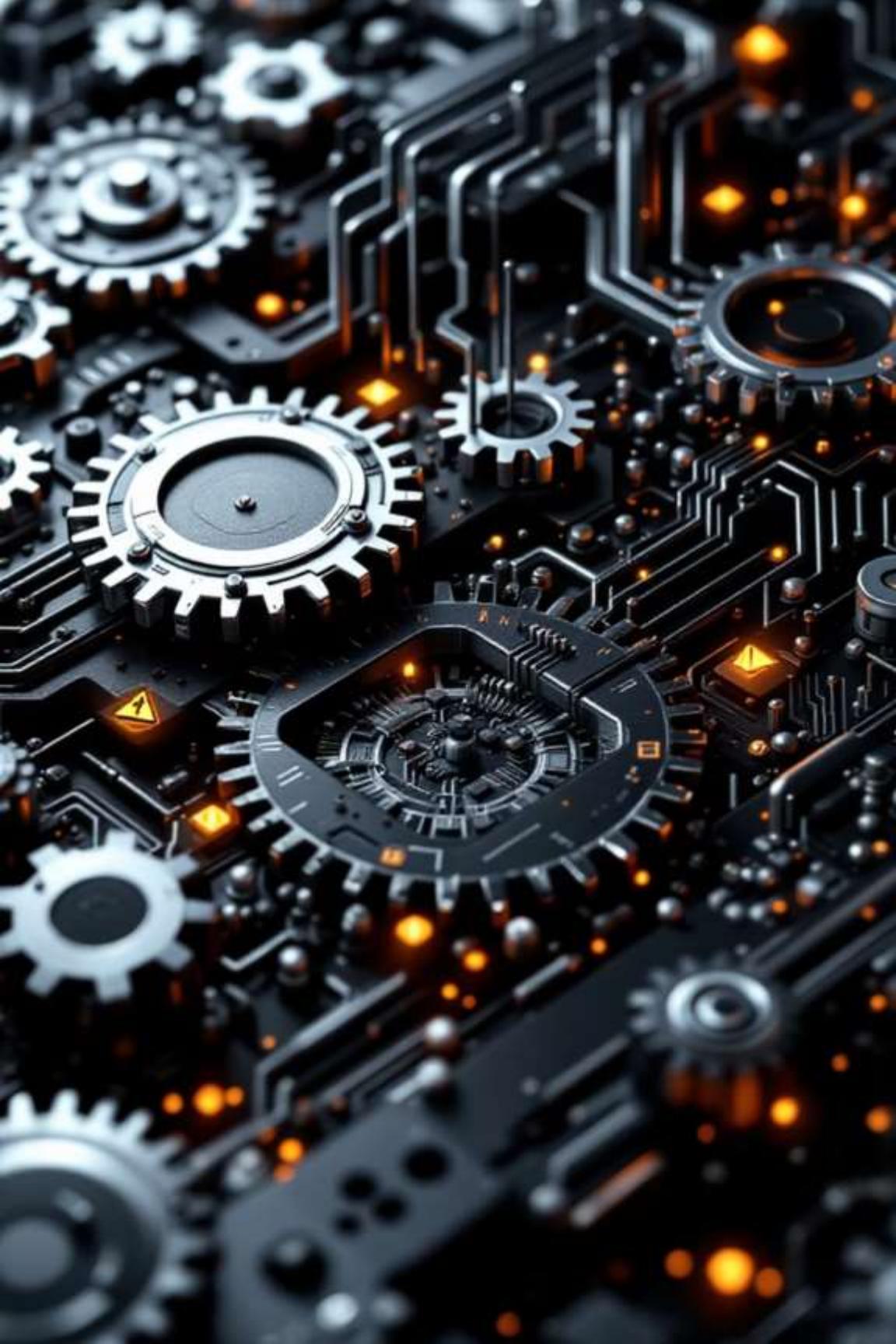


Student-Centric Learning

The system delivers personalized exercises, practice problems, and provides real-time feedback to students, adapting to their learning pace.

Data-Driven Decisions

Schools can leverage granular data insights from TeachSmart to make informed decisions, continually optimizing learning strategies and student outcomes.



Challenges and Ethical Considerations

Data Privacy & Accuracy

Ensuring the security of student data and verifying the precision of AI assessments are paramount concerns in educational AI.

Bias & Fairness

It is critical to develop AI tools with transparency and fairness to prevent algorithmic bias, ensuring equitable opportunities for all students.

Responsible Use

Comprehensive training for educators is essential, enabling them to integrate and utilize AI technologies effectively and ethically within the classroom.

Future of AI in Smart Learning

Immersive Learning

Seamless integration of AI with Virtual (VR) and Augmented Reality (AR) will create highly immersive and interactive educational experiences.

Predictive Analytics

Advanced AI-driven analytics will forecast academic challenges and potential student dropouts, allowing for proactive support.

Global Accessibility

AI-powered platforms will democratize access to high-quality education, reaching underserved populations worldwide.

Conclusion: Embracing AI to Empower Teachers and Students

AI serves as a powerful tool to enhance—not replace—the invaluable human creativity and profound connection that define great teaching.

Smart learning systems, enriched by artificial intelligence, hold the immense potential to elevate educational quality and foster greater equity across all learning environments.

Your capstone project represents a unique opportunity to contribute directly to shaping the future of learning, innovating how knowledge is shared and acquired for generations to come.



Capstone Project:

<https://gamma.app/docs/The-Impact-of-Social-Media-on-Education-Capstone-Project-Name-Dev-oj3c55oqisg67wn>

Certificate of Completion

AI Literacy Skills 101

Devansh Lakra

has successfully completed the AI Literacy Skills 101 Course, demonstrating a foundational understanding of how to use AI thoughtfully, ethically, and creatively in teaching and learning.



Modules covered:

Introduction to AI

Critical Thinking with AI

Creativity with AI

Computational Thinking

Self and Social Awareness

Collaborating with AI

Communicating AI Use

Problem Solving with AI



AI FOR ALL

Devansh Lakra

AI AWARE 2025

intel®