Alliance School of Advanced Computing

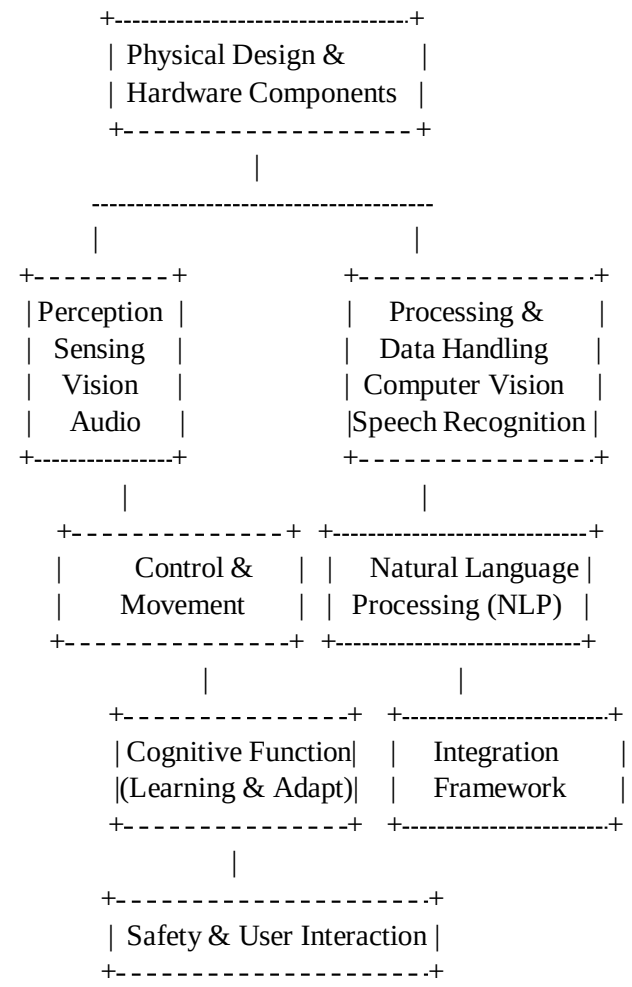
Department of Computer Science and Engineering

Class Assignment-1

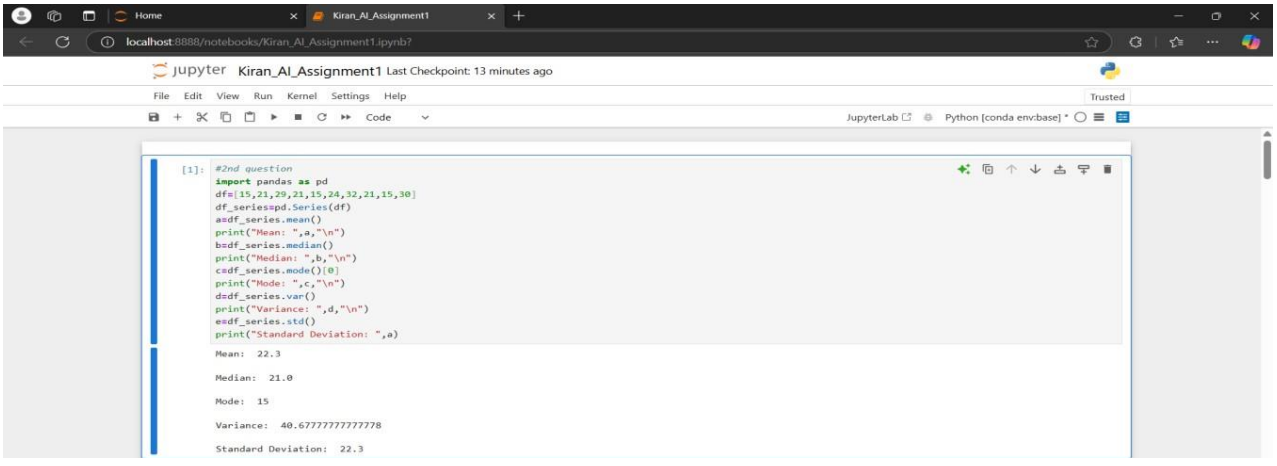
Course Code: 5CS1025
Course Title: Artificial Intelligence
Semester: 04
Class: AIML-E
Name: KIRAN R
Reg no: 2023BCSE07AED409
Github: https://github.com/KiranR2005/Kiran_AI_Assignment1.git

2024-25

1. Imagine you are tasked with designing a humanoid robot to assist in a home or office environment. The robot must be capable of interacting with people by **talking** and **listening**, **walking** to different locations, **seeing** and recognizing objects, and **learning** from its surroundings to adapt its behavior. What technologies, tools, and frameworks would you need to build such a robot? Give as flow chart



2. Calculate and interpret mean, median, mode, variance and standard deviation for a given dataset. Data = [15,21,29,21,15,24,32,21,15,30]

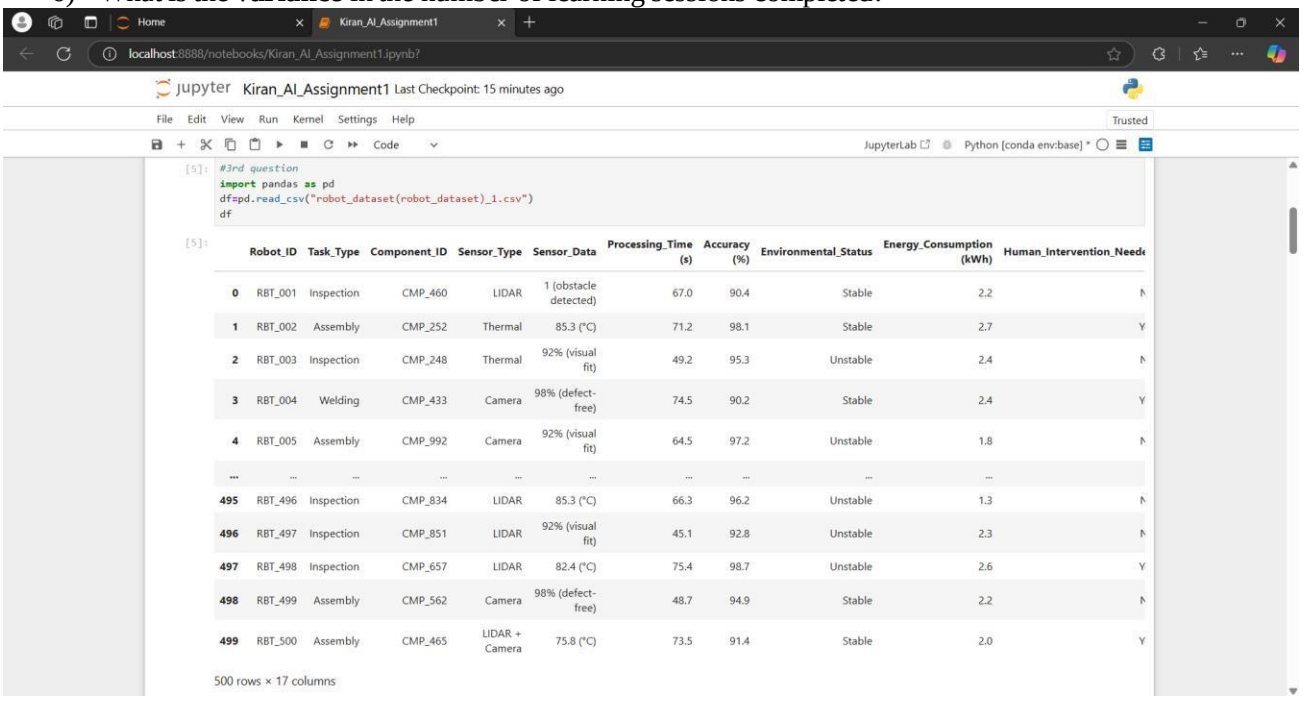


3. You are analyzing a dataset that captures the daily performance and activity of a humanoid robot in a simulated environment. The dataset link robot_dataset(robot_dataset)_1.csv includes the following attributes

Interaction_Count: Number of conversations the robot had daily.
Steps_Walked: Total steps taken each day.
Objects_Recognized: Number of objects successfully identified by the robot.
Learning_Sessions: Number of learning tasks completed.
Energy_Consumption (kWh): Daily energy usage of robots.

Perform Basic Statistical Operations:

1) What is the **average (mean)** number of conversations the robot has daily?
2) Find the **total steps walked** by the robot over a given period.
3) Determine the **maximum and minimum energy consumption** in the dataset.
4) Calculate the **correlation** between the number of steps walked and energy consumption.
5) Analyze the **distribution** of objects recognized daily (e.g., histogram or box plot).
6) What is the **variance** in the number of learning sessions completed?



	Robot_ID	Task_Type	Component_ID	Sensor_Type	Sensor_Data	Processing_Time (s)	Accuracy (%)	Environmental_Status	Energy_Consumption (kWh)	Human_Intervention_Needed
0	RBT_001	Inspection	CMP_460	LIDAR	1 (obstacle detected)	67.0	90.4	Stable	2.2	N
1	RBT_002	Assembly	CMP_252	Thermal	85.3 (°C)	71.2	98.1	Stable	2.7	Y
2	RBT_003	Inspection	CMP_248	Thermal	92% (visual fit)	49.2	95.3	Unstable	2.4	N
3	RBT_004	Welding	CMP_433	Camera	98% (defect-free)	74.5	90.2	Stable	2.4	Y
4	RBT_005	Assembly	CMP_992	Camera	92% (visual fit)	64.5	97.2	Unstable	1.8	N
...	...	...	...	...	...	...	...	...	...	...
495	RBT_496	Inspection	CMP_834	LIDAR	85.3 (°C)	66.3	96.2	Unstable	1.3	N
496	RBT_497	Inspection	CMP_851	LIDAR	92% (visual fit)	45.1	92.8	Unstable	2.3	N
497	RBT_498	Inspection	CMP_657	LIDAR	82.4 (°C)	75.4	98.7	Unstable	2.6	Y
498	RBT_499	Assembly	CMP_562	Camera	98% (defect-free)	48.7	94.9	Stable	2.2	N
499	RBT_500	Assembly	CMP_465	LIDAR + Camera	75.8 (°C)	73.5	91.4	Stable	2.0	Y

Home | Kiran_AI_Assignment1

localhost:8888/notebooks/Kiran_AI_Assignment1.ipynb?

jupyter Kiran_AI_Assignment1 Last Checkpoint: 16 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code | JupyterLab | Python [conda env:base] *

[9]:
```
df.describe()
```

[9]:

	Processing_Time (s)	Accuracy (%)	Energy_Consumption (kWh)	Interaction_Count	Steps_Walked	Objects_Recognized	Learning_Sessions	Energy_Consumption (kWh).1
count	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000
mean	59.937000	94.920600	2.014000	5.510000	28.758000	4.650000	35.960000	236.250000
std	11.688808	2.900949	0.582789	2.486416	11.094757	2.521948	19.797532	146.681008
min	40.100000	90.000000	1.000000	0.000000	0.000000	0.000000	10.000000	100.000000
25%	49.675000	92.500000	1.500000	3.000000	22.000000	2.750000	20.000000	119.500000
50%	59.600000	94.900000	2.000000	6.000000	29.000000	5.000000	31.000000	181.500000
75%	70.000000	97.500000	2.500000	7.000000	36.000000	7.000000	49.000000	320.250000
max	80.000000	99.900000	3.000000	12.000000	49.000000	9.000000	91.000000	654.000000

[11]:
```
correlation = df['Steps_Walked'].corr(df['Energy_Consumption (kWh)']) #correlation between the number of steps walked and energy consumption.
print("Correlation between steps and energy consumption:", correlation)
```

```
Correlation between steps and energy consumption: 0.0015478137393314443
```

Home | Kiran_AI_Assignment1

localhost:8888/notebooks/Kiran_AI_Assignment1.ipynb?

jupyter Kiran_AI_Assignment1 Last Checkpoint: 17 minutes ago

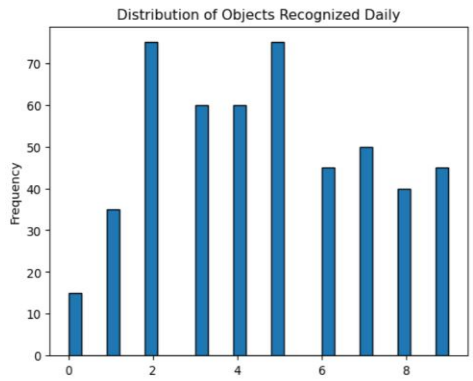
File Edit View Run Kernel Settings Help

Trusted

Code | JupyterLab | Python [conda env:base] *

[13]:
```
#distribution of objects recognized daily (Histogram plot).
df['Objects_Recognized'].plot(kind='hist', bins=30, edgecolor='black', title='Distribution of Objects Recognized Daily')
```

[13]:
```
<Axes: title={'center': 'Distribution of Objects Recognized Daily'}, ylabel='Frequency'>
```



[15]:
```
#variance in the number of learning sessions.
variance = df['Learning_Sessions'].var()
print("Variance in the number of learning sessions:", variance)
```

```
Variance in the number of learning sessions: 391.9422845691385
```

Home | Kiran_AI_Assignment1

localhost:8888/notebooks/Kiran_AI_Assignment1.ipynb?

jupyter Kiran_AI_Assignment1 Last Checkpoint: 16 minutes ago

File Edit View Run Kernel Settings Help

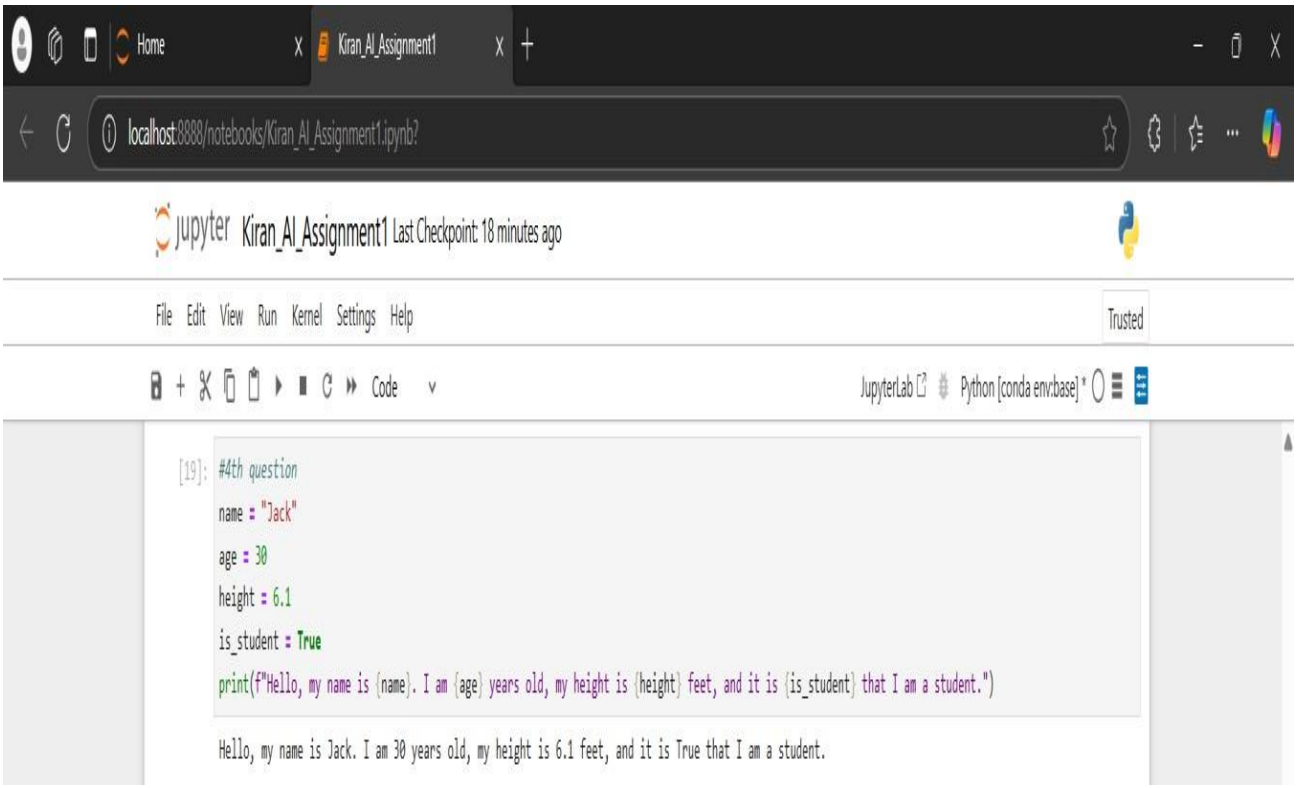
Trusted

Code | JupyterLab | Python [conda env:base] *

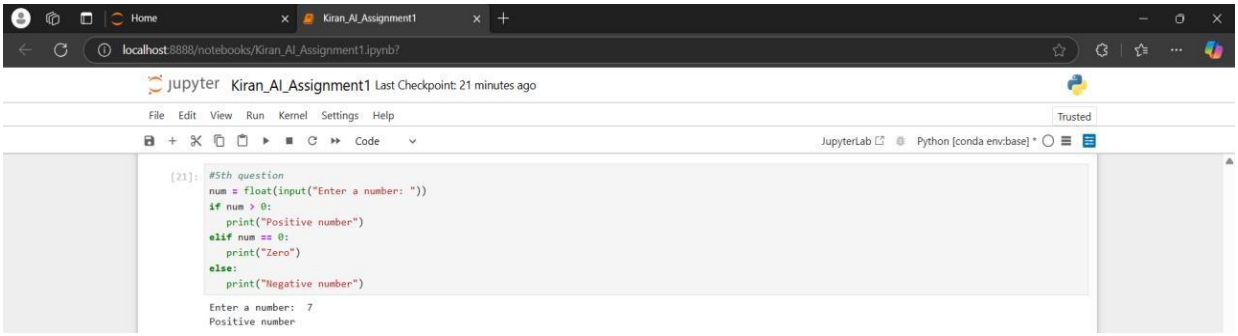
[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 17 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Robot_ID                    500 non-null    object
 1   Task_Type                   500 non-null    object
 2   Component_ID                500 non-null    object
 3   Sensor_Type                 500 non-null    object
 4   Sensor_Data                 500 non-null    object
 5   Processing_Time (s)         500 non-null    float64
 6   Accuracy (%)                500 non-null    float64
 7   Environmental_Status        500 non-null    object
 8   Energy_Consumption (kWh)    500 non-null    float64
 9   Human_Intervention_Needed   500 non-null    object
 10  Obstacle_Detected           500 non-null    object
 11  Defect_Detected             500 non-null    object
 12  Interaction_Count           500 non-null    int64
 13  Steps_Walked                500 non-null    int64
 14  Objects_Recognized          500 non-null    int64
 15  Learning_Sessions           500 non-null    int64
 16  Energy_Consumption (kWh).1  500 non-null    int64
dtypes: float64(3), int64(5), object(9)
memory usage: 66.5+ KB
```

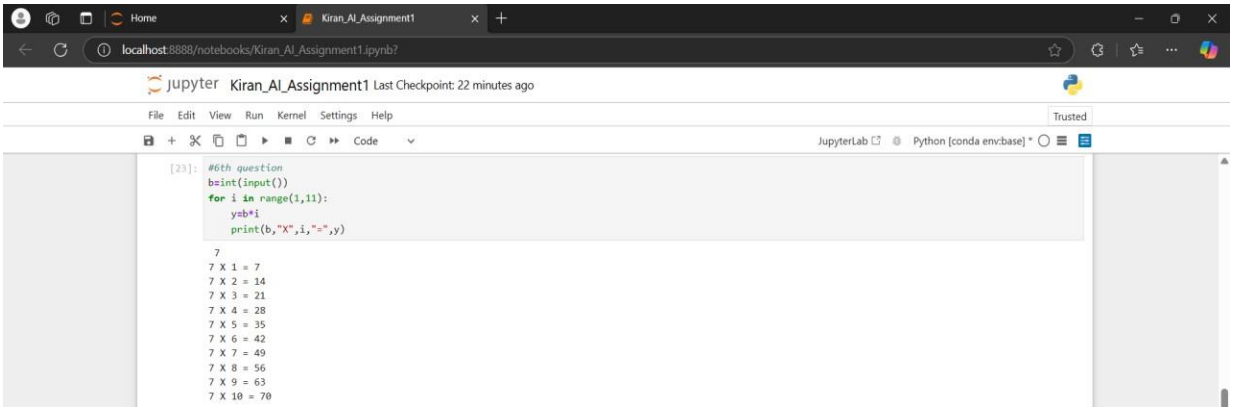
4. Write a Python program that declares variables of different data types (e.g., string, integer, float, and boolean). Output the variables in a sentence format using print() and f-strings.



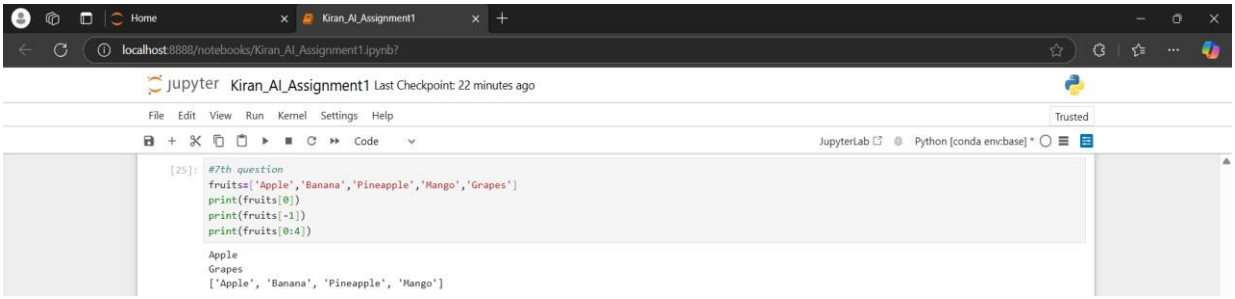
5. Write a Python program that takes an integer input and checks whether the number is positive, negative, or zero using conditional statements (if-else).



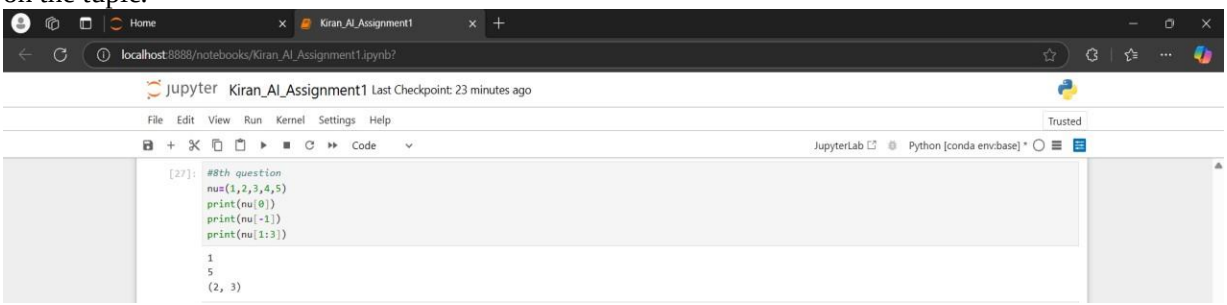
6. Write a Python program that takes a number as input and prints the multiplication table for that number (from 1 to 10).



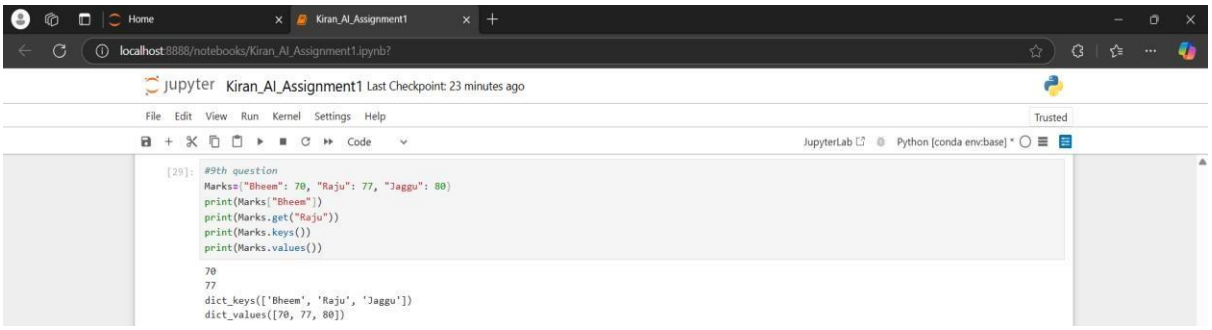
7. Create a Python list that contains the names of 5 different fruits. Perform the given operations on the list.



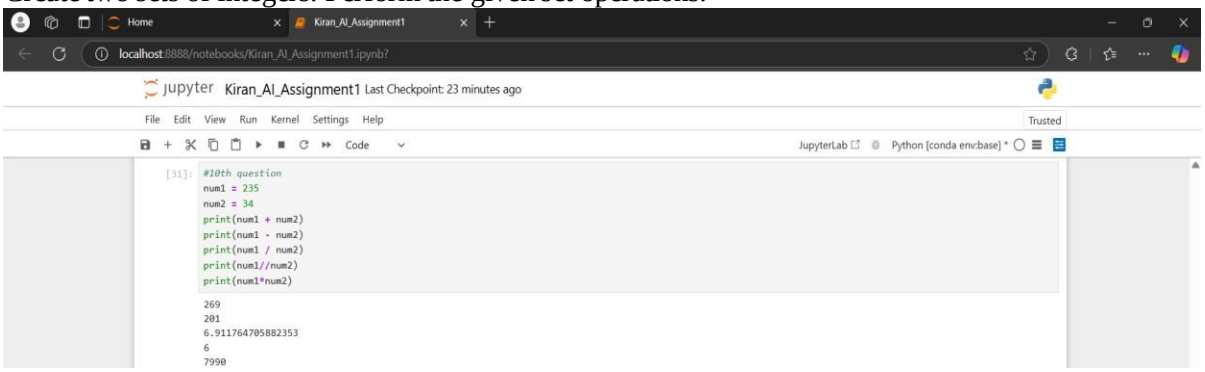
8. Write a Python program that creates a tuple containing 5 numbers. Perform the given operations on the tuple.



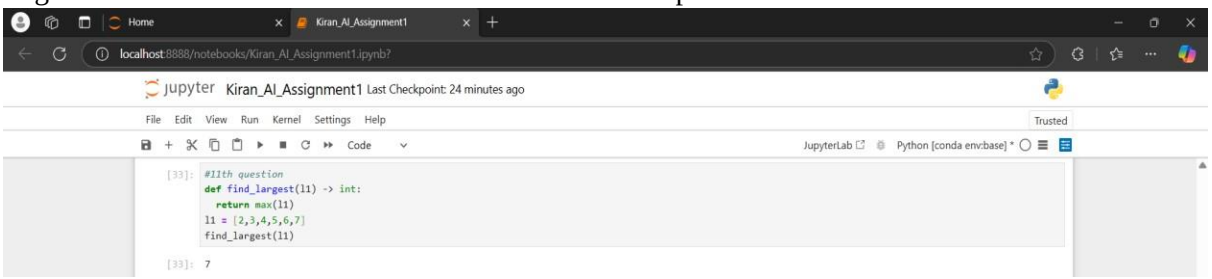
9. Create a dictionary that stores the names of 3 students as keys and their marks in mathematics as values. Perform the given operations.



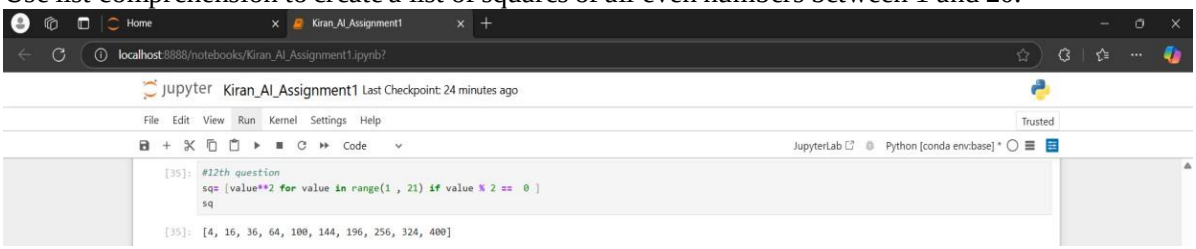
10. Create two sets of integers. Perform the given set operations.



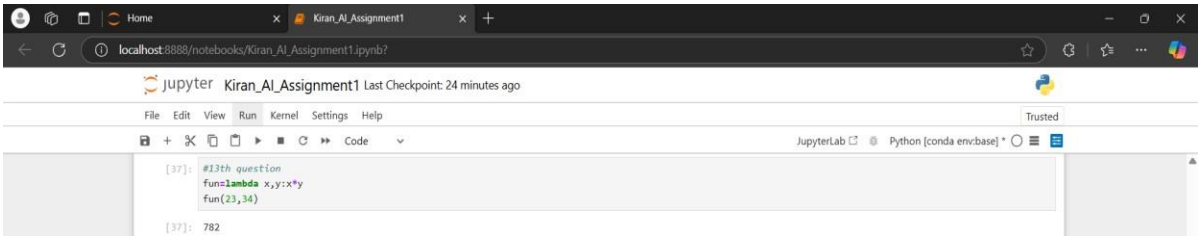
11. Write a Python function called find_largest() that takes a list of numbers as input and returns the largest number from the list. Test the function with a sample list.



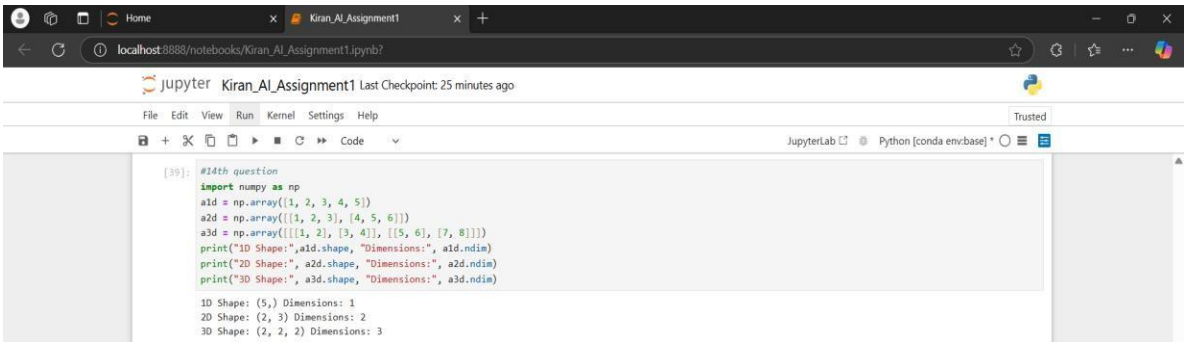
12. Use list comprehension to create a list of squares of all even numbers between 1 and 20.



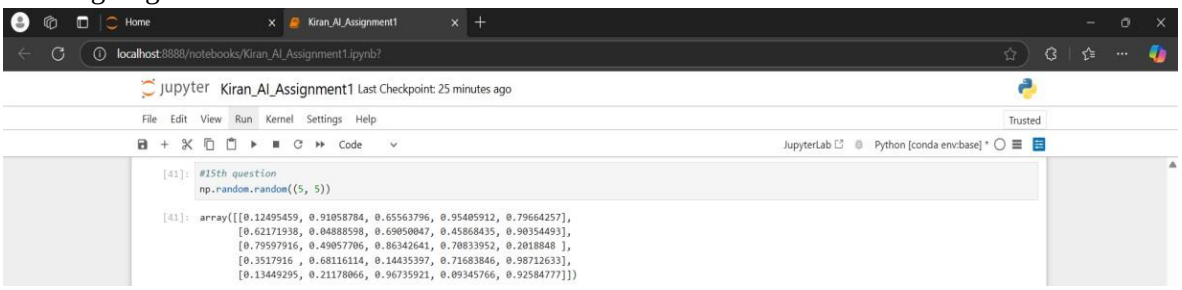
13. Write a Python script that uses a lambda function to calculate the product of two numbers provided by the user.



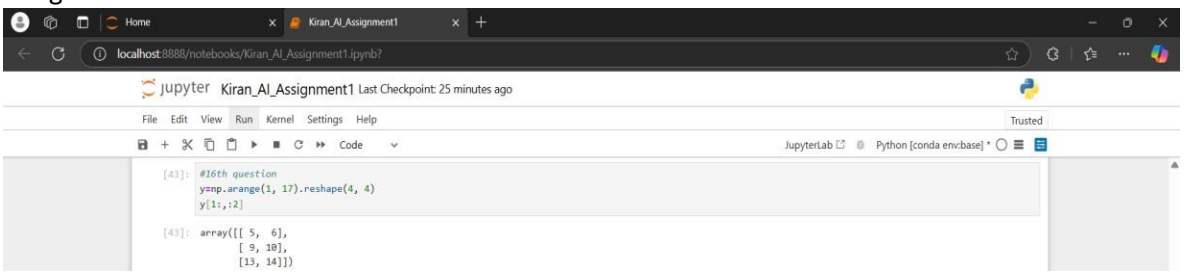
14. Write a Python program to create a one-dimensional, two-dimensional, and three-dimensional NumPy array. Print the shape and dimensions of each array.



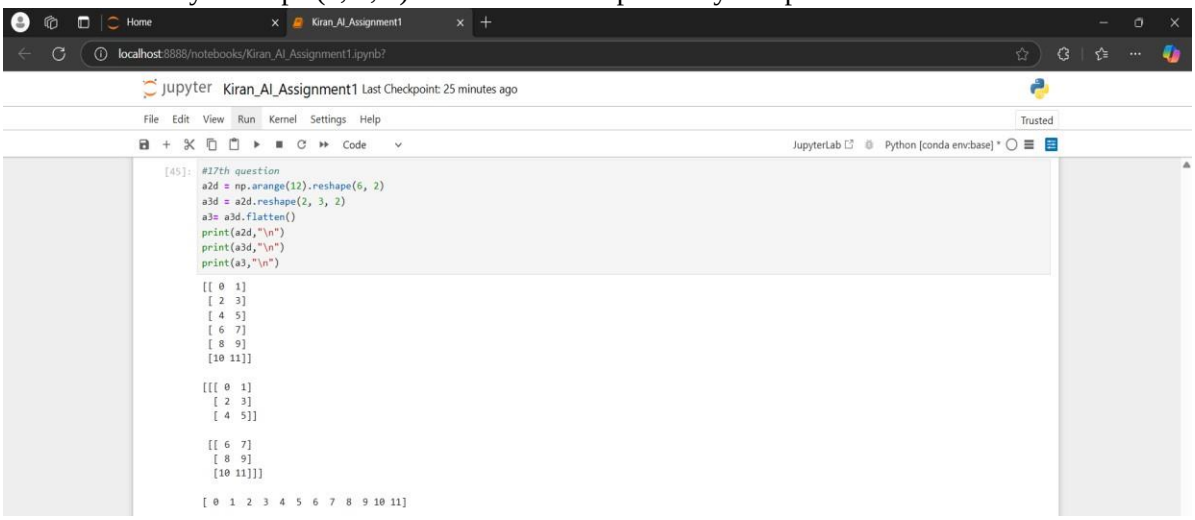
15. Write a Python program to create a 5x5 NumPy array of random integers and Perform array indexing as given.



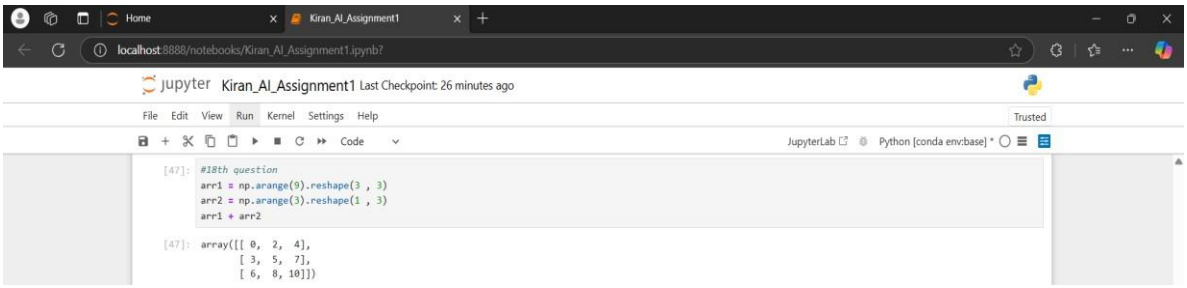
16. create a NumPy array of shape (4, 4) containing numbers from 1 to 16. Use slicing to extract for the given conditions



17. Write a Python program that creates a 2D array of shape (6, 2) using np.arange() and then reshapes it into a 3D array of shape (2, 3, 2). Flatten the reshaped array and print the result.



18. Write a Python program to demonstrate broadcasting. Create an array of shape (3, 3) and add a one-dimensional array of shape (1, 3) to it using broadcasting.



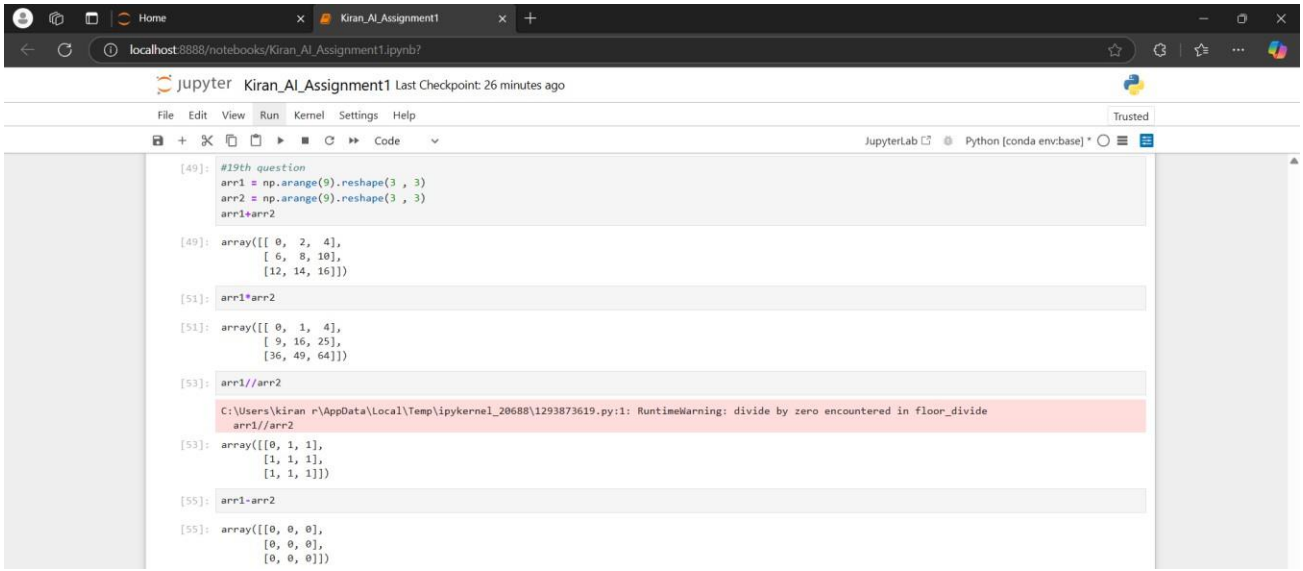
19. Create two NumPy arrays of the same shape, A and B. Perform the following arithmetic operations:

Element-wise addition.

Element-wise subtraction.

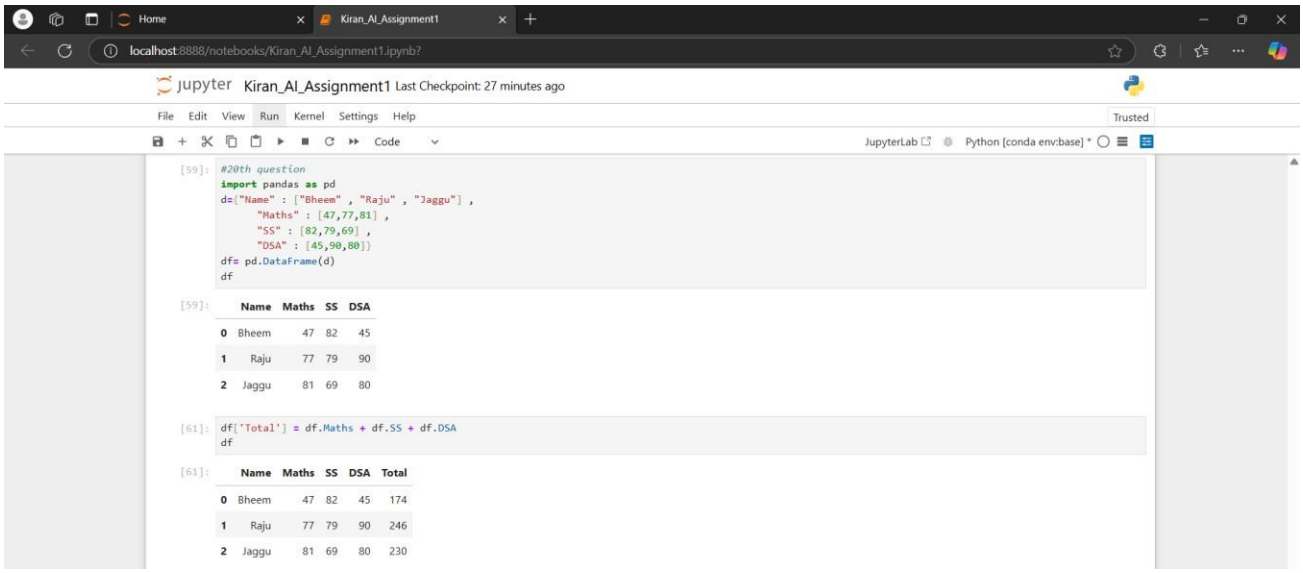
Element-wise multiplication.

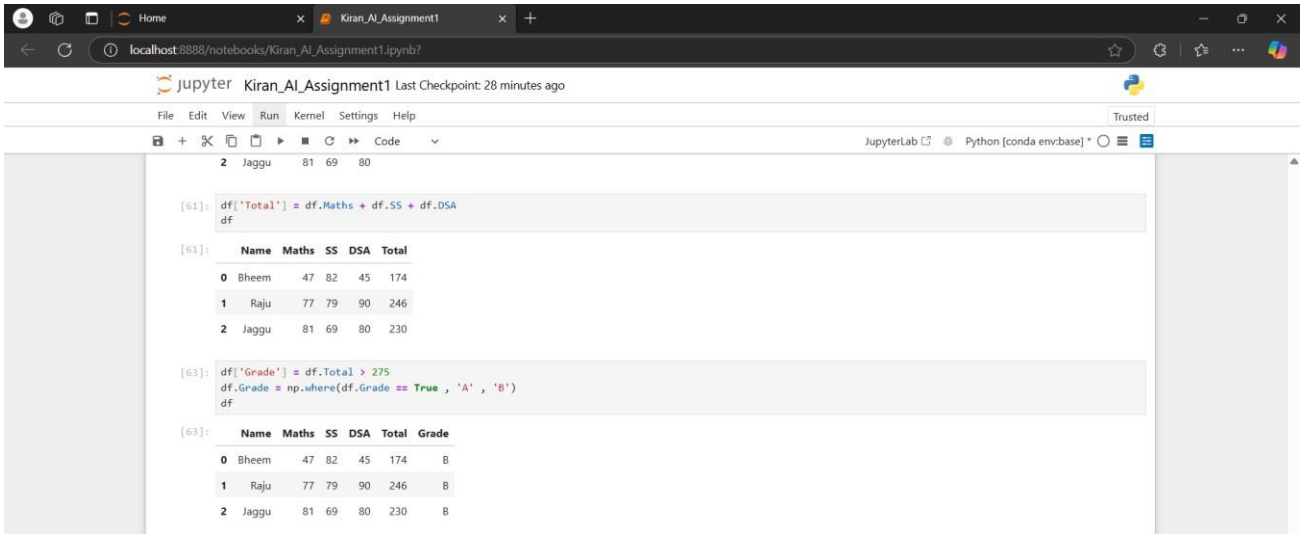
Element-wise division.



20. Create a Pandas DataFrame with the given Name and marks of 3 courses:

Add a new column named 'Total' that represents the sum of all the courses. Add 'Grade' based on the values of the 'Total'. Print the updated DataFrame with the new 'Total' and 'Grade' column.


Home
Kiran_AI_Assignment1
localhost:8888/notebooks/Kiran_AI_Assignment1.ipynb?
jupyter Kiran_AI_Assignment1 Last Checkpoint: 27 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *
[59]:
#20th question
import pandas as pd
d={"Name" : ["Bheem" , "Raju" , "Jaggu"] ,
"Maths" : [47,77,81] ,
"SS" : [82,79,69] ,
"DSA" : [45,90,80]}
df= pd.DataFrame(d)
df
[59]:
Name Maths SS DSA
0 Bheem 47 82 45
1 Raju 77 79 90
2 Jaggu 81 69 80
[61]:
df['Total'] = df.Maths + df.SS + df.DSA
df
[61]:
Name Maths SS DSA Total
0 Bheem 47 82 45 174
1 Raju 77 79 90 246
2 Jaggu 81 69 80 230


Home
Kiran_AI_Assignment1
localhost:8888/notebooks/Kiran_AI_Assignment1.ipynb?
jupyter Kiran_AI_Assignment1 Last Checkpoint: 28 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *
2 Jaggu 81 69 80
[61]:
df['Total'] = df.Maths + df.SS + df.DSA
df
[61]:
Name Maths SS DSA Total
0 Bheem 47 82 45 174
1 Raju 77 79 90 246
2 Jaggu 81 69 80 230
[63]:
df['Grade'] = df.Total > 275
df.Grade = np.where(df.Grade == True , 'A' , 'B')
df
[63]:
Name Maths SS DSA Total Grade
0 Bheem 47 82 45 174 B
1 Raju 77 79 90 246 B
2 Jaggu 81 69 80 230 B