GL BAJAJ INSTITUTE OF TECHNOLOGY AND MANAGEMENT

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Session—2025-26.

Subject—Python Programming Lab
Subject code—BCC302

Submitted to:

Mr Shikhar Prateek Pandey

Submitted by: Aditya
Roll no - 2401920100035
CSE-2A

Unit — 1: Introduction in Python

Question-1: Write a Python program that;

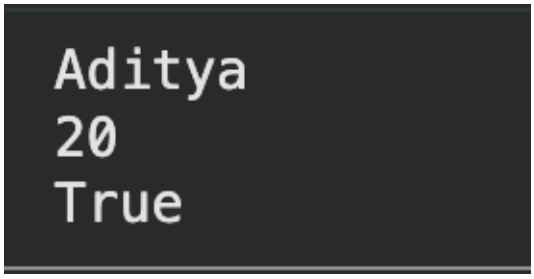
a) Stores your name, age , and whether you're a student in separate variables.

```
name = "Aditya"
age = 20 # Born in 2005
is_student = True
```

b) Prints them all with clear labels.

```
print(name)
print(age)
print(is_student)
```

Output :



Question-2:Write a Python program to:

Take two numbers, perform and print addition, subtraction, multiplication, division and modulus.

Code :

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))

print("Addition:", a + b)
print("Subtraction:", a - b)
print("Multiplication:", a * b)
print("Division:", a / b)
print("Floor Division:", a // b)
print("Modulus:", a % b)
```

Output :

```
Enter first number: 7
Enter second number: 2
Addition: 9
Subtraction: 5
Multiplication: 14
Division: 3.5
Floor Division: 3
Modulus: 1
```

Unit -2 : PYTHON PROGRAM FLOW CONTROL CONDITIONAL BLOCKS

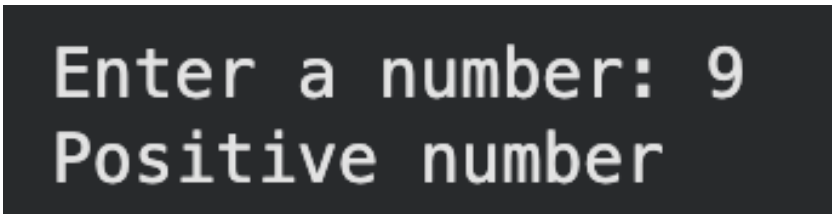
Question-1:Write a Python program to:

Asks the user to enter a number and uses if, elif, and else to check if the number is positive, negative, or zero.

Code :

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

if num > 0:
    print("Positive number")
elif num < 0:
    print("Negative number")
else:
    print("Zero")
```

Output :



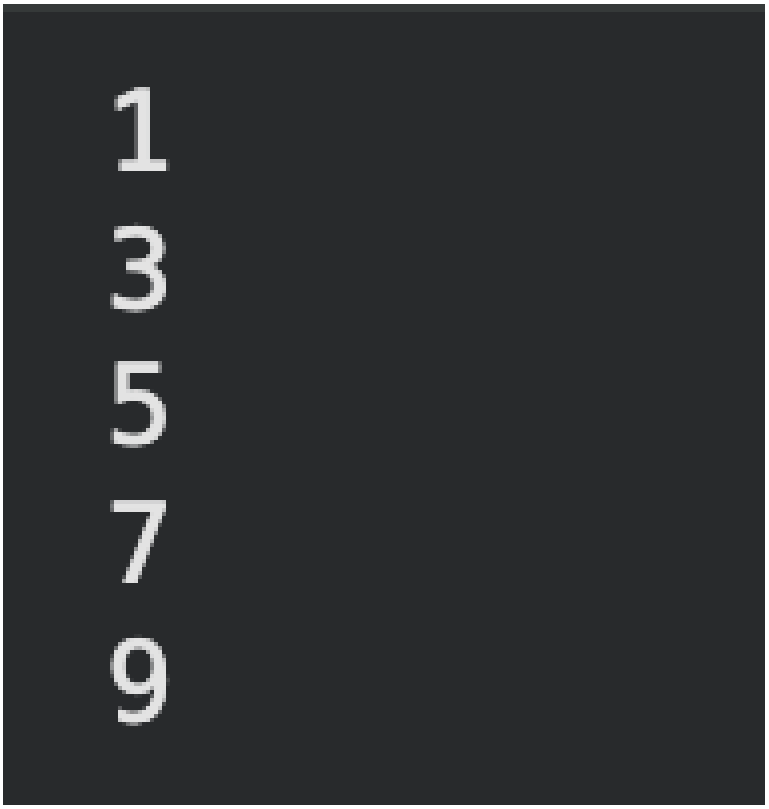
Question-2 : Write a Python program to:

Uses a for loop with range to print numbers from 1 to 10, skipping even numbers using continue.

Code :

```
for i in range(1, 11):
    if i % 2 == 0:
        continue
    print(i)
```

Output :



Unit-3:Python Complex data types

Write a Python program to:

a). Accept a sentence and demonstrate string manipulations (length, slicing, case conversion, finding substrings, etc.).

Code :

```
text = input("Enter a sentence: ")

# length
print("Length:", len(text))

# slicing
print("First 5 characters:", text[:5])
print("Last 5 characters:", text[-5:])

# case conversion
print("Uppercase:", text.upper())
print("Lowercase:", text.lower())
print("Title case:", text.title())

# finding substring
word = input("Enter a word to search: ")
pos = text.find(word)

if pos != -1:
    print("Substring found at index:", pos)
else:
    print("Substring not found")
```

Output :

```
Enter a sentence: The cat jumped over the small wooden fence.
Length: 43
First 5 characters: The c
Last 5 characters: ence.
Uppercase: THE CAT JUMPED OVER THE SMALL WOODEN FENCE.
Lowercase: the cat jumped over the small wooden fence.
Title case: The Cat Jumped Over The Small Wooden Fence.
Enter a word to search: cat
Substring found at index: 4
```

b). Create a list of numbers/items and perform slicing, appending, inserting, and removing items

Code :

```
items = [10, 20, 30, 40, 50]

# slicing
print("First 3 items:", items[:3])
print("Last 2 items:", items[-2:])

# appending
items.append(60)
print("After append:", items)

# inserting
items.insert(2, 25)   # insert 25 at index 2
print("After insert:", items)

# removing
items.remove(40)      # remove value 40
print("After remove:", items)
```

Output :

```
First 3 items: [10, 20, 30]
Last 2 items: [40, 50]
After append: [10, 20, 30, 40, 50, 60]
After insert: [10, 20, 25, 30, 40, 50, 60]
After remove: [10, 20, 25, 30, 50, 60]
```

c). Define a tuple with at least 5 elements and demonstrate indexing and slicing

Code :

```
t = (5, 10, 15, 20, 25)

# indexing
print("First element:", t[0])
print("Third element:", t[2])

# slicing
print("First three elements:", t[:3])
print("Last two elements:", t[-2:])
```

Output :

```
First element: 5
Third element: 15
First three elements: (5, 10, 15)
Last two elements: (20, 25)
```

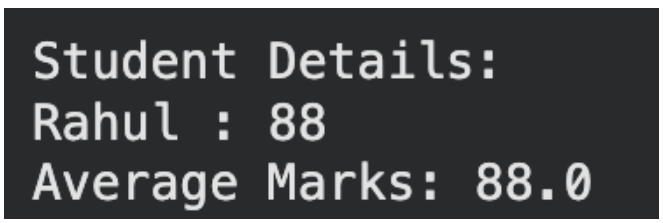
Write a Python program using dictionaries and functions to store and process student information:

- **Store student names and marks in a dictionary.**
- **Perform dictionary operations (add, update, delete, display).**
- **Define and use functions (e.g., add student, calculate average marks, display details).**

Code :

```
students = {}

def add_student(name, marks):
    students[name] = marks

def update_student(name, marks):
    students[name] = marks

def delete_student(name):
    if name in students:
        del students[name]

def display_students():
    print("Student Details:")
    for name, marks in students.items():
        print(name, ":", marks)

def average_marks():
    if len(students) == 0:
        return 0
    return sum(students.values()) / len(students)

# Example usage
add_student("Aditya", 90)
add_student("Rahul", 85)
update_student("Rahul", 88)
delete_student("Aditya")
display_students()

print("Average Marks:", average_marks())
```

Output :



Unit-4:Python File Operations

Question-1: Write a Python program to open a text file and demonstrate he use of read(), read line(), and read lines() functions.

Code :

```
# Opening a text file in read mode
file = open("sample.txt", "r")

print("Using read():")
print(file.read())

# Move cursor back to start
file.seek(0)

print("Using readline():")
print(file.readline())

# Move cursor back to start
file.seek(0)

print("\nUsing readlines():")
print(file.readlines())

file.close()
```

Output:

```
Using read():
Hello
World
This is sample
Using readline():
Hello

Using readlines():
['World\n', 'This is sample']
```

Question-2: Write a Python program to create a text file and write multiple lines into it using write() and write lines().

Code :

```
# Opening a text file in read mode
file = open("sample.txt", "r")
print("Using read():")
print(file.read())
# Move cursor back to start
file.seek(0)
print("Using readline():")
print(file.readline())
# Move cursor back to start
file.seek(0)
print("\nUsing readlines():")
print(file.readlines())
file.close()
```

Output :

```
File created and written successfully!
```

Unit-5: Python Packages

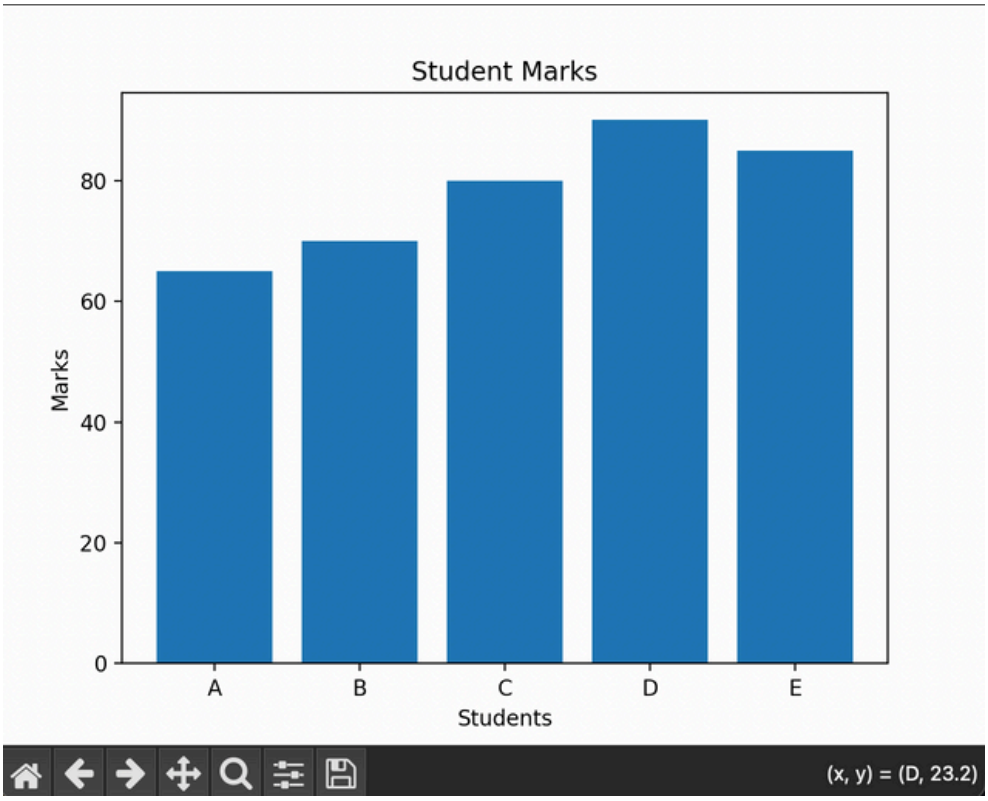
Question-1:Data Visualization, Analysis using Matplotlib, Pandas and NumPy.

Code :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# ----- NUMPY -----
# Create a NumPy array
data = np.array([10, 20, 30, 40, 50])

print("NumPy Array:", data)
print("Mean:", np.mean(data))
print("Sum:", np.sum(data))

# ----- PANDAS -----
# Convert to DataFrame
df = pd.DataFrame({
    "Marks": [65, 70, 80, 90, 85],
    "Students": ["A", "B", "C", "D", "E"]
})

print("\nPandas DataFrame:")
print(df)

print("\nAverage Marks:", df["Marks"].mean())

# ----- MATPLOTLIB -----
# Plotting bar graph
plt.bar(df["Students"], df["Marks"])
plt.title("Student Marks")
plt.xlabel("Students")
plt.ylabel("Marks")
plt.show()
```

Output :

```
Matplotlib is building the font cache; this may take a moment.
NumPy Array: [10 20 30 40 50]
Mean: 30.0
Sum: 150

Pandas DataFrame:
   Marks Students
0     65        A
1     70        B
2     80        C
3     90        D
4     85        E

Average Marks: 78.0
```



Question-2:Implement Tkinter and PythonProgramming for calculators.

Code :

```
import tkinter as tk

# Button click function
def click(btn):
    current = entry.get()
    entry.delete(0, tk.END)
    entry.insert(0, current + btn)

def clear():
    entry.delete(0, tk.END)

def calculate():
    try:
        result = eval(entry.get())
        entry.delete(0, tk.END)
        entry.insert(0, str(result))
    except:
        entry.delete(0, tk.END)
        entry.insert(0, "Error")

# Main window
root = tk.Tk()
root.title("Simple Calculator")

entry = tk.Entry(root, width=20, font=("Arial", 20), borderwidth=5)
entry.grid(row=0, column=0, columnspan=4)

# Buttons layout
buttons = [
    "7","8","9","/",
    "4","5","6","*",
    "1","2","3","-",
    "0",".","=","+"
]

row = 1
col = 0

for btn in buttons:
    if btn == "=":
        tk.Button(root, text=btn, width=5, height=2,
                  command=calculate).grid(row=row, column=col)
    else:
        tk.Button(root, text=btn, width=5, height=2,
                  command=lambda b=btn: click(b)).grid(row=row, column=col)

    col += 1
    if col == 4:
        col = 0
        row += 1

tk.Button(root, text="C", width=5, height=2, command=clear).grid(row=row, column=0)

root.mainloop()
```

Output :