

```
# Aim: Write a Program to make simple calculator using if statements
# Name: Sidra Solkar
# UIN:231P087 Roll No.: 43
```

```
print("*****Simple Calculator*****")
while True:
    # Displaying the menu
    print("\nMENU")
    print("1. Addition")
    print("2. Subtraction")
    print("3. Multiplication")
    print("4. Division")
    print("5. Exit")
    # Taking user input for the operation choice
    choice = int(input("Enter your choice: "))
    if choice >= 1 and choice <= 4:
        # Prompting for two numbers
        print("Enter two numbers: ")
        num1 = float(input("First number: "))
        num2 = float(input("Second number: "))
        # Performing the selected operation
        if choice == 1:
            res = num1 + num2
            print("Result = ", res)
        elif choice == 2:
            res = num1 - num2
            print("Result = ", res)
        elif choice == 3:
            res = num1 * num2
            print("Result = ", res)
        elif choice == 4:
            if num2 != 0:
                res = num1 / num2
                print("Result = ", res)
            else:
                print("Error! Division by zero.")
    elif choice == 5:
        print("Exiting the calculator. Goodbye!")
        print("Sidra Solkar \n43 \nSEComps(A)")
        break
    else:
        print("Wrong input...!! Please enter a valid choice.")
```

```
# Aim:Write a menu-driven program in Python to implement list operations
# Name: Sidra Solkar
# UIN:231P087 Roll No.: 43
```

```
print("*****Menu-Driven List Operations*****")
# Initial Lists
list1 = ["python", "list", "1974", "2005", "1983"]
```

```

list2 = ["Sidra", "Solkar", "Batminton", "player", "2025", "February"]
while True:
    print("\nMENU")
    print("1. Create and Display Lists")
    print("2. Find Length of a List")
    print("3. Check if an Element is in the List")
    print("4. Concatenate Two Lists")
    print("5. Replace an Element in a List")
    print("6. Delete an Element from a List")
    print("7. Work with Nested Lists")
    print("8. Exit")
    # User choice
    choice = input("Enter your choice: ")
    if choice == "1":
        print("\ni. Create and Display Lists")
        print("list1:", list1)
        print("list2:", list2)
        print("list1[1:4]:", list1[1:4])
        print("list1[1:]:", list1[1:])
        print("list1[0]:", list1[0])
        print("list1 * 2:", list1 * 2)
        print("list1 + list2:", list1 + list2)
    elif choice == "2":
        print("\nii. Find Length of a List")
        print("Length of list1:", len(list1))
        print("Length of list2:", len(list2))
    elif choice == "3":
        print("\niii. Check if an Element is in the List")
        element = input("Enter the element to check: ")
        if element in list1:
            print(f"'{element}' is in list1.")
        else:
            print(f"'{element}' is not in list1.")
    elif choice == "4":
        print("\niv. Concatenate Two Lists")
        result_list = list1 + list2
        print("Result of concatenation:", result_list)
    elif choice == "5":
        print("\nv. Replace an Element in a List")
        print("Original list1:", list1)
        try:
            index = int(input("Enter the index to replace: "))
            new_value = input("Enter the new value: ")
            if 0 <= index < len(list1):
                list1[index] = new_value
                print("Updated list1:", list1)
            else:
                print("Invalid index!")
        except ValueError:
            print("Invalid input! Please enter a valid index.")

```

```

elif choice == "6":
    print("\nvi. Delete an Element from a List")
    print("Original list1:", list1)
    try:
        index = int(input("Enter the index to delete: "))
        if 0 <= index < len(list1):
            del list1[index]
            print("Updated list1:", list1)
        else:
            print("Invalid index!")
    except ValueError:
        print("Invalid input! Please enter a valid index.")
elif choice == "7":
    print("\nvii. Work with Nested Lists")
    books_list = [("C#", 896), ("Java", 750), ("Python", 1269)]
    while True:
        print("\nNested List Operations:")
        print("1. Show Books List")
        print("2. Add a Book")
        print("3. Back to Main Menu")
        nested_choice = input("Enter your choice: ")
        if nested_choice == "1":
            print("\nBook\t\t\tPrice")
            for book, price in books_list:
                print(f"{book}\t\t\t{price}")
        elif nested_choice == "2":
            book_name = input("\nEnter the name of the book: ")
            try:
                book_price = int(input("Enter its price: "))
                books_list.append((book_name, book_price))
                print("Book added successfully!")
            except ValueError:
                print("Invalid price! Please enter a numeric value.")
        elif nested_choice == "3":
            break
        else:
            print("Invalid choice! Please try again.")
elif choice == "8":
    print("Exiting the program. Goodbye!")
    print("Sidra Solkar \n43 \nSEComps(A)")
    break
else:
    print("Invalid choice! Please select a valid option.")

```

```

# Aim: Write a menu-driven program in Python to implement tuple operations
# Name: Sidra Solkar
# UIN: 231P087 Roll No.: 43

```

```

print("*****Tuple Operations*****")
tuple1 = ("Sara", "Sidra", "Shifa", "Joya")

```

```

tuple2 = ("Helicopter", "Aeroplane", "Aircraft")
while True:
    print("\nMenu:")
    print("1. Create tuple")
    print("2. Display tuple")
    print("3. Find length of tuple")
    print("4. Check if element is present in tuple")
    print("5. Concatenate tuples")
    print("6. Delete tuple")
    print("7. Count occurrences of an element in tuple")
    print("8. Exit")
    choice = input("\nEnter your choice (1-8): ")
    if choice == '1':
        tuple1 = ("Sara", "Sidra", "Shifa", "Joya")
        print("\nTuple created.")
    elif choice == '2':
        print("\nDisplaying the tuple:")
        print(tuple1)
        print("\nIndividual elements:")
        for item in tuple1 :
            print(item)
    elif choice == '3':
        length_of_tuple = len(tuple1)
        print(f"\nLength of the tuple: {length_of_tuple}")
    elif choice == '4':
        element = input("\nEnter the element to check: ")
        if element in tuple1:
            print(f"'{element}' is present in the tuple.")
        else:
            print(f"'{element}' is not present in the tuple.")
    elif choice == '5':
        concatenated_tuple = tuple1 + tuple2
        print("\nConcatenated tuple:")
        print(concatenated_tuple)
    elif choice == '6':
        del tuple1
        print("\nTuple has been deleted.")
    elif choice == '7':
        tuple1 = ("Sara", "Sidra", "Shifa", "Joya")
        element = input("\nEnter the element to count: ")
        count = tuple1.count(element)
        print(f"\n'{element}' appears {count} times in the tuple.")
    elif choice == '8':
        print("\nExiting the program.")
        print("Sidra Solkar \n43 \nSEComps(A)")
        break
    else:
        print("\nInvalid choice. Please enter a number between 1 and 8.")

```

Aim: Write a menu-driven program in Python to implement set operations

```
# Name: Sidra Solkar
# UIN:231P087 Roll No.: 43
```

```
print("*****Set Operations*****")
A, B = set(), set()
while True:
    print("\n***** Set Operations Menu *****")
    print("1. Create Sets")
    print("2. Union, Intersection, Difference, Symmetric Difference")
    print("3. Modify Set")
    print("4. Remove Elements from Set")
    print("5. Use Pop and Clear")
    print("6. Check if an Item Exists in a Set")
    print("7. Exit")

    choice = input("Enter your choice (1-7): ")
    if choice == '1':
        A = {0, 2, 4, 6, 8}
        B = {11, 2, 13, 4, 15}
        print("Set A:", A)
        print("Set B:", B)
    elif choice == '2':
        print("Union of A and B:", A | B)
        print("Intersection of A and B:", A & B)
        print("Difference of A and B (A - B):", A - B)
        print("Symmetric Difference of A and B:", A ^ B)
    elif choice == '3':
        set1 = {1, 3}
        print("Initial Set:", set1)
        set1.add(2)
        print("After Adding 2:", set1)
        set1.update([2, 3, 4])
        print("After Adding Multiple Elements:", set1)
        set1.update([4, 5], {1, 6, 8})
        print("After Adding List and Set:", set1)
    elif choice == '4':
        set1 = {1, 3, 4, 5, 6}
        print("Initial Set:", set1)
        set1.discard(4)
        print("After Discarding 4:", set1)
        set1.remove(6)
        print("After Removing 6:", set1)
        set1.discard(2) # No error if not present
        print("After Discarding 2:", set1)
    elif choice == '5':
        set1 = set("HelloWorld")
        print("Initial Set:", set1)
        print("Popped Element:", set1.pop())
        print("Set after pop:", set1)
        set1.clear()
```

```

        print("Set after clear:", set1)
    elif choice == '6':
        set1 = set("apple")
        print("Set:", set1)
        print("Is 'a' in the set?", 'a' in set1)
        print("Is 'p' not in the set?", 'p' not in set1)
    elif choice == '7':
        print("Exiting Program. Goodbye!")
        print("Sidra Solkar \n43 \nSEComps(A)")
        break
    else:
        print("Invalid choice! Please enter a number between 1 and 7.")

```

Aim: Write a menu-driven program in Python to implement Dictionary operations
 # Name: Sidra Solkar
 # UIN: 231P087 Roll No.: 43

```

print("*****Dictionary Operations*****")
while True:
    print("\n***** Dictionary Operations Menu *****")
    print("1. Create Dictionary")
    print("2. Access Elements from Dictionary")
    print("3. Change or Add Elements in Dictionary")
    print("4. Delete or Remove Elements from Dictionary")
    print("5. Find Length and Perform Sorting")
    print("6. Exit")

    choice = input("Enter your choice (1-6): ")
    if choice == '1':
        dict1 = {}
        print("Empty Dictionary:", dict1)
        dict1 = {1: 'aeroplane', 2: 'Boeing'}
        print("Dictionary with Integer Keys:", dict1)
        dict1 = {'name': 'Sidra', 1: [2, 4, 3]}
        print("Dictionary with Mixed Keys:", dict1)
        dict1 = dict({1: 'aeroplane', 2: 'Boeing'})
        print("Using dict():", dict1)
        dict1 = dict([(1, 'aeroplane'), (2, 'Boeing')])
        print("From Sequence as Pairs:", dict1)
    elif choice == '2':
        dict1 = {'name': 'Sidra', 'age': 18}
        print("Dictionary:", dict1)
        print("Access name:", dict1['name'])
        print("Access age:", dict1.get('age'))
    elif choice == '3':
        dict1 = {'name': 'Sidra', 'age': 18}
        dict1['age'] = 19
        print("Updated Age:", dict1)
        dict1['address'] = 'Mumbai'
        print("After Adding Address:", dict1)

```

```

elif choice == '4':
    dict2 = {1:1, 2:8, 3:27, 4:64, 5:125}
    print("Dictionary Before Removal:", dict2)
    print("Popped Element (Key 4):", dict2.pop(4))
    print("Dictionary After pop(4):", dict2)
    print("Popped Arbitrary Item:", dict2.popitem())
    print("Dictionary After popitem():", dict2 )
    del dict2[3]
    print("Dictionary After Deleting Key 3:", dict2)
    dict2.clear()
    print("Dictionary After clear():", dict2)
elif choice == '5':
    dict2= {1: 1, 3: 27, 5:125, 7:256, 9:743}
    print("Length of Dictionary:", len(dict2))
    print("Sorted Keys:", sorted(dict2))
elif choice == '6':
    print("Exiting Program. Goodbye!")
    print("Sidra Solkar \n43 \nSE Comps(A)")
    break
else:
    print("Invalid choice! Please enter a number between 1 and 6.")

```

```

# postLab
# Aim: Write a program in python to compute factorial of a number.
# Name: Sidra Solkar
# UIN:231P087 Roll No.: 43

```

```

print("*****Factorial Of a Number*****")
num = int(input("Enter a non-negative integer: "))
if num < 0:
    print("Factorial is not defined for negative numbers.")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    factorial = 1
    for i in range(1, num + 1):
        factorial *= i
    print(f"The factorial of {num} is {factorial}")
    print("\nSidra Solkar \n43 \nSE Copms(A)")

```

```

#Aim:Write a program in python to display the following pattern.

```

```

#
# #
# # #
# # # #
# Name: Sidra Solkar
# UIN:231P087 Roll No.: 43

```

```

print("*****Pattern*****")
rows = 4

```

```
for i in range(1, rows + 1):
    print("# " * i)
print("\nSidra Solkar \n43 \nSEComps(A)")
```

```
# Aim:WAP Recursive function to calculate sum of a number from 0 to n.
# Name: Sidra Solkar
# UIN:231P087 Roll No.: 43
```

```
print("*****Recursive Function*****")
def recur_sum(n):
    if n == 0:
        return 0
    else:
        return n + recur_sum(n-1)
n = int (input("Enter a number:"))
if n < 0:
    print("Enter a positive number")
else:
    print("The sum is",recur_sum(n))
print("\nSidra Solkar \n43 \nSEComps(A)")
```

```
# Aim: Write a Program in python Define a function that accepts roll number and
returns whether the student is present or absent.
# Name: Sidra Solkar
# UIN:231P087 Roll No.: 43
```

```
print("*****Attendance*****")
def check_attendance(roll_number, attendance_list):
    if roll_number in attendance_list:
        return "Present"
    else:
        return "Absent"
attendance_list = [1,2,3,4,5,6,8,10,11,15,19,30,31,33,34,35,40,43]
roll_number = int(input("Enter the roll number of the student: "))
status = check_attendance(roll_number, attendance_list)
print(f"The student with roll number {roll_number} is {status}.")
print("\nSidra Solkar \n43 \nSEComps(A)")
```

```
# PostLab
# Aim: Write a Program to find largest between three number using functions.
# Name: Sidra Solkar
# UIN:231P087 Roll No.: 43
```

```
print("*****Largest Number*****")
def find_largest(num1, num2, num3):
    if num1 >= num2 and num1 >= num3:
        return num1
    elif num2 >= num1 and num2 >= num3:
        return num2
    else:
```



```

        return num3
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
num3 = int(input("Enter the third number: "))
largest = find_largest(num1, num2, num3)
print(f"The largest number between {num1},{num2} and {num3} is {largest}.")
print("\nSidra Solkar \n43 \nSEComps(A)")

```

Aim: Write a Program in python to implement Bubble sort using functions.
Name: Sidra Solkar
UIN:231P087 Roll No.: 43

```

def bubble_sort(arr):
    n = len(arr)

    for i in range(n):

        swapped = False
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
                swapped = True

        if not swapped:
            break

arr = list(map(int, input("Enter numbers separated by spaces: ").split()))
bubble_sort(arr)
print("Sorted array is:", arr)
print("\nSidra Solkar \n43 \nSEComps(A)")

```

Aim: Write a Program to implement Inheritance in python.
Name: Sidra Solkar
UIN:231P087 Roll No.: 43

```

class Person:
    name=""
    age=0
    def __init__(self,name,age):
        self.name=name
        self.age=age
    def display(self):
        print("Name :",self.name)
        print("Age :",self.age)

class Teacher(Person):
    exp=0
    r_area=""
    def __init__(self,name,age,exp,r_area):
        Person.__init__(self,name,age)

```

```

        self.exp=exp
        self.r_area=r_area
    def displayData(self):
        Person.display(self)
        print("Experience :",self.exp)
        print("Research Area :",self.r_area)
class Student(Person):
    course=""
    marks=0
    def __init__(self,name,age,course,marks):
        Person.__init__(self,name,age)
        self.course=course
        self.marks=marks
    def displayData(self):
        Person.display(self)
        print("Course :",self.course)
        print("Marks :",self.marks)
print("*****TEACHER*****")
teacher_name = input("Enter Teacher's Name: ")
teacher_age = int(input("Enter Teacher's Age: "))
teacher_exp = int(input("Enter Teacher's Years Of Experience: "))
teacher_r_area = input("Enter Teacher's Research Area: ")

print("*****STUDENT*****")
student_name = input("Enter Student's Name: ")
student_age = int(input("Enter Student's Age: "))
student_course = input("Enter Student's Course: ")
student_marks = float(input("Enter Student's Marks: "))

print("*****TEACHER*****")
T = Teacher(teacher_name, teacher_age, teacher_exp, teacher_r_area)
T.displayData()
print("*****STUDENT*****")
S = Student(student_name, student_age, student_course, student_marks)
S.displayData()
print("\nSidra Solkar \n43 \nSEComps(A)")

```

Aim: Write a Program in python to implement Multiple Inheritance.
 # Name: Sidra Solkar
 # UIN:231P087 Roll No.: 43

```

print("*****Multiple Inheritance*****")
class Employee:
    def __init__(self, emp_id, emp_name):
        self.emp_id = emp_id
        self.emp_name = emp_name
    def set_emp_id(self, emp_id):
        self.emp_id = emp_id
    def get_emp_id(self):
        return self.emp_id

```

```

    def set_emp_name(self, emp_name):
        self.emp_name = emp_name
    def get_emp_name(self):
        return self.emp_name
class Student:
    def __init__(self, student_id, student_name, student_college):
        self.student_id = student_id
        self.student_name = student_name
        self.student_college = student_college
    def set_student_id(self, student_id):
        self.student_id = student_id
    def get_student_id(self):
        return self.student_id
    def set_student_name(self, student_name):
        self.student_name = student_name
    def get_student_name(self):
        return self.student_name
    def set_student_college(self, student_college):
        self.student_college = student_college
    def get_student_college(self):
        return self.student_college
class Intern(Employee, Student):
    def __init__(self, emp_id, emp_name, student_id, student_name, student_college,
period):
        Employee.__init__(self, emp_id, emp_name)
        Student.__init__(self, student_id, student_name, student_college)
        self.period = period
    def set_period(self, period):
        self.period = period
    def get_period(self):
        return self.period
    def display_intern_details(self):
        print(f"Employee ID: {self.get_emp_id()}")
        print(f"Employee Name: {self.get_emp_name()}")
        print(f"Student ID: {self.get_student_id()}")
        print(f"Student Name: {self.get_student_name()}")
        print(f"Student College: {self.get_student_college()}")
        print(f"Internship period: {self.get_period()} months")
intern = Intern(emp_id="8765", emp_name="Shifa", student_id="231P087",
student_name="Sidra Solkar", student_college="RCOE", period=6)
intern.display_intern_details()
print("\nSidra Solkar \n43 \nSEComps(A)")

```

postLab

Aim: Write a program in python to calculate volume of sphere using multilevel inheritance.

Name: Sidra Solkar

UIN:231P087 Roll No.: 43

```
print("*****Multilevel Inheritance*****")
```

```

class Circle:
    def __init__(self):
        self.radius = 0
    def accept_radius(self):
        self.radius = float(input("Enter the radius of the sphere: "))
class Area(Circle):
    def __init__(self):
        super().__init__()
    def calculate_area(self):
        area = 3.14 * (self.radius ** 2)
        print(f"Area of the circle: {area:.2f}")
class Volume(Area):
    def __init__(self):
        super().__init__()
    def calculate_volume(self):
        volume = (4/3) * 3.14 * (self.radius ** 3)
        print(f"Volume of the sphere: {volume:.2f}")
sphere = Volume()
sphere.accept_radius()
sphere.calculate_area()
sphere.calculate_volume()
print("\nSidra Solkar \n43 \nSEComps(A)")

```

Aim: Write a program in python to calculate volume of sphere using multilevel inheritance demonstrating method overriding.

Name: Sidra Solkar

UIN:231P087 Roll No.: 43

```

print("*****Method overriding*****")
class Circle:
    def __init__(self):
        self.radius = 0
    def accept_radius(self):
        self.radius = float(input("Enter the radius of the sphere: "))
class Area(Circle):
    def __init__(self):
        super().__init__()
    def accept_radius(self):
        print("In Area class: Calculating area of the circle.")
        super().accept_radius()
    def calculate_area(self):
        area = 3.14159 * (self.radius ** 2)
        print(f"Area of the circle: {area:.2f}")
class Volume(Area):
    def __init__(self):
        super().__init__()
    def accept_radius(self):
        print("In Volume class: Calculating volume of the sphere.")
        super().accept_radius()
    def calculate_volume(self):

```

```
        volume = (4/3) * 3.14159 * (self.radius ** 3)
        print(f"Volume of the sphere: {volume:.2f}")
sphere = Volume()
sphere.accept_radius()
sphere.calculate_area()
sphere.calculate_volume()
print("\nSidra Solkar \n43 \nSEComps(A)")
```