Al Assisted Coding Ass-6.3

B. Sushanth

2403A52L11—B50

Task 1: Classes (Student Class)

Scenario:

You are developing a simple student information management module.

Prompt and code:

```
#Task-1
""" prompt: Write a Python class named Student with attributes for name, roll number, and branch. Include a method to display the student's details.
    Create an instance of the Student class and call the method to display the details."""

class Student :
    def __init__(self,name,roll_number,branch) :
        self.name = name
        self.roll_number = roll_number
        self.branch = branch
    def display_details(self) :
        print(f"Name: {self.name}")
        print(f"Roll Number: {self.roll_number}")
        print(f"Branch: {self.branch}")

name = input("Enter student name: ")
roll_number = input("Enter student roll number: ")
branch = input("Enter student branch: ")

student1 = Student(name, roll_number, branch)
student1.display_details()
```

Output:

```
PS C:\Users\saiki\AppData\Local\Programs\Microsoft VS Code> & C:\Python313\python.exe c:/Users/saiki/OneDrive/Desktop/python/My_codes/6.3.py
Enter student name: Sushanth
Enter student roll number: 2111
Enter student branch: CSE
Name: Sushanth
Roll Number: 2111
Branch: CSE
```

Task 2: Loops (Multiples of a Number)

Prompt and Code:

```
#Task-2
""" prompt: Write a Python program that takes a number as input and prints the multiples of that number from 1 to 10 using both for and while loops."""

number = int(input("Enter a number :"))
print("Multiples of the number using for loop :")
for i in range(1,11) :
    print(number * i)
print("Multiples of the number using while loop :")
i = 1
while i <= 10 :
    print(number * i)
    i += 1
```

Output:

```
PS C:\Users\saiki\AppData\Local\Programs\Microsoft VS Code> & C:\Python313\python.exe c:/Users/saiki/OneDrive/Desktop/python/My_codes/6.3.py
Enter a number :4
Multiples of the number using for loop :
4
8
12
16
20
24
28
32
36
40
Multiples of the number using while loop :
4
8
12
16
20
24
28
32
36
40
```

Task 3: Conditional Statements (Age Classification)

Prompt and Code:

```
#Task-3
""" prompt: Write a Python program that takes a person's age as input and categorizes them as a child (0-12),
    |   |   teenager (13-19), adult (20-59), or senior citizen (60 and above) using if-elif-else statements."""

age = int(input("Enter your age :"))
if age < 13 :
    print("You are a child.")
elif age < 20 :
    print("You are a teenager.")
elif age < 60 :
    print("You are an adult.")
else :
    print("You are a senior citizen.")
```

Output:

```
PS C:\Users\saiki\AppData\Local\Programs\Microsoft VS Code> & C:\Python313\python.exe c:/Users/saiki/OneDrive/Desktop/python/My_codes/6.3.py
Enter your age :20
You are an adult.
```

Task 4: For and While Loops (Sum of First n Numbers)

Prompt and code:

```
#Task-4
""" prompt: Write a Python program that calculates the sum of the first n natural numbers using a for loop, a while loop, and a formula.
    |   |   Take n as input from the user and display the results from all three methods."""

def sum_of_n_for(n) :
    total = 0
    for i in range(1, n+1) :
    |   total += i
    return total
def sum_of_n_while(n) :
    total = 0
    i = 1
    while i <= n :
    |   total += i
    |   i += 1
    return total
def sum_of_n_formula(n) :
    return n * (n + 1) // 2
n = int(input("Enter a number :"))
print(f"Sum of first {n} numbers using for loop: {sum_of_n_for(n)}")
print(f"Sum of first {n} numbers using while loop: {sum_of_n_while(n)}")
print(f"Sum of first {n} numbers using formula: {sum_of_n_formula(n)}")
```

Output:

```
PS C:\Users\saiki\AppData\Local\Programs\Microsoft VS Code> & C:\Python313\python.exe c:/Users/saiki/OneDrive/Desktop/python/My_codes/6.3.py
Enter a number :5
Sum of first 5 numbers using for loop: 15
Sum of first 5 numbers using while loop: 15
Sum of first 5 numbers using formula: 15
```

Task 5: Classes (Bank Account Class)

Prompt and Code:

```
#Task-5
""" prompt: Write a Python class named BankAccount with attributes for account holder name and balance.
        Include methods for depositing money,withdrawing money,and checking the balance.Create an instance of the BankAccount class
        and demonstrate the functionality of each method by performing a series of transactions and displaying the balance after each transaction. """

class BankAccount :
    def __init__(self, account_holder, balance=0) :
        self.account_holder = account_holder
        self.balance = balance
    def deposit(self, amount) :
        if amount > 0 :
            self.balance += amount
            print(f"Deposited: {amount}. New Balance: {self.balance}")
        else :
            print("Deposit amount must be positive.")
    def withdraw(self, amount) :
        if amount > 0 and amount <= self.balance :
            self.balance -= amount
            print(f"Withdrew: {amount}. New Balance: {self.balance}")
        else :
            print("Invalid withdrawal amount or insufficient funds.")
    def check_balance(self) :
        print(f"Current Balance: {self.balance}")
account_holder = input("Enter account holder name: ")
initial_balance = float(input("Enter initial balance: "))
account = BankAccount(account_holder, initial_balance)
account.check_balance()
deposit_amount = float(input("Enter amount to deposit: "))
account.deposit(deposit_amount)
withdraw_amount = float(input("Enter amount to withdraw: "))
account.withdraw(withdraw_amount)
account.check_balance()
```

Output:

```
PS C:\Users\saiki\AppData\Local\Programs\Microsoft VS Code> & C:\Python313\python.exe c:/Users/saiki/OneDrive/Desktop/python/My_codes/6.3.py
Enter account holder name: sushanth
Enter initial balance: 500
Current Balance: 500.0
Enter amount to deposit: 200
Deposited: 200.0. New Balance: 700.0
Enter amount to withdraw: 500
Withdrew: 500.0. New Balance: 200.0
Current Balance: 200.0
```