# 9.3-Assignment

NAME:PRANEETH BANDA

2303A51711

BATCH-20

## TASK-1:

```
Task-1:
Function: sum_even_odd(numbers)
Purpose: Separates a list of integers into even and odd numbers,
         then computes and returns the sum of each group as a tuple.
Parameters:
    numbers (list): A list of integers to process
Returns:
    tuple: (sum_of_evens, sum_of_odds) where:
        sum_of_evens (int): The sum of all even numbers in the list
        sum_of_odds (int): The sum of all odd numbers in the list
Examples:
    sum_even_odd([1, 2, 3, 4, 5, 6]) returns (12, 9)
    sum_even_odd([10, 15, 20, 25]) returns (30, 40)
    sum_even_odd([]) returns (0, 0)
Notes:
    Empty lists return (0, 0)
    Negative numbers are handled correctly (-2 is even, -3 is odd)
"""
def sum_even_odd(numbers):
    """
    Computes the sum of even and odd numbers separately.
    """
    sum_even = sum(num for num in numbers if num % 2 == 0)
    sum_odd = sum(num for num in numbers if num % 2 != 0)
    return sum_even, sum_odd
"""
Test the function
"""
if __name__ == "__main__":
    test_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
    even_sum, odd_sum = sum_even_odd(test_list)
    print(f"Sum of even numbers: {even_sum}")
    print(f"Sum of odd numbers: {odd_sum}")
```

## OUTPUT:

```
assignment.py"
Sum of even numbers: 30
Sum of odd numbers: 25
```

## TASK-2:

```
Task-2
Class: sru_student
Purpose: Manages student information and fees at SRU
Attributes:
    name (str): Student's name
    roll_no (int): Student's roll number
    hostel_status (str): Yes/No indicating if student lives in hostel
    fee (int): Total accumulated fee amount (initialized to 0)
Methods:
    __init__(name, roll_no, hostel_status): Initialize student details
    fee_update(amount): Add given amount to existing fee
    display_details(): Print all student information and current fee
"""
class sru_student:
    """
    Constructor to initialize student details
    """
    def __init__(self, name, roll_no, hostel_status):
        """
        Assign student name, roll number, hostel status and initialize fee.
        """
        self.name = name
        self.roll_no = roll_no
        self.hostel_status = hostel_status
        self.fee = 0
    """
    Method to update student fee
    """
    def fee_update(self, amount):
        """
        Add given amount to existing fee.
        """
        self.fee += amount
    """
```

```
    Method to display student details
    """
    def display_details(self):
        """
        Print all student details and fee.
        """
        print(f"Name: {self.name}")
        print(f"Roll Number: {self.roll_no}")
        print(f"Hostel Status: {self.hostel_status}")
        print(f"Fee: {self.fee}")
"""
Test the class
"""
if __name__ == "__main__":
    student = sru_student("Alice", 101, "Yes")
    student.fee_update(5000)
    student.display_details()
```

# OUTPUT:

```
assignment.py"
Name: john
Roll Number: 102
Hostel Status: Yes
Fee: 15000
```

# TASK-3:

```
Task-3:
Calculator Module
=================
A lightweight calculator module providing basic arithmetic operations.
This module is designed for use across multiple projects and provides
clean, well-documented functions for common mathematical calculations.
Functions:
    add : Add two numbers
    subtract : Subtract two numbers
    multiply : Multiply two numbers
    divide : Divide two numbers with error handling
Examples:
    add(5, 3) -> 8
    divide(10, 2) -> 5.0
"""
def add(a, b):
    """
    Add two numbers.
    Parameters:
        a (float): The first number.
        b (float): The second number.
    Returns:
        float: The sum of a and b.
    """
    return a + b
def subtract(a, b):
    """
    Subtract two numbers.
    Parameters:
        a (float): The minuend.
        b (float): The subtrahend.
    Returns:
        float: The difference of a and b.
```

```python
def multiply(a, b):
    """
    Multiply two numbers.
    Parameters:
        a (float): The first factor.
        b (float): The second factor.
    Returns:
        float: The product of a and b.
    """
    return a * b
def divide(a, b):
    """
    Divide two numbers.
    Parameters:
        a (float): The dividend.
        b (float): The divisor.
    Returns:
        float: The quotient of a and b.
    Raises:
        ValueError: If b is zero.
    """
    if b == 0:
        raise ValueError("Cannot divide by zero")
    return a / b
"""
Test the calculator functions
"""
if __name__ == "__main__":
    print(f"Add: {add(10, 5)}")
    print(f"Subtract: {subtract(10, 5)}")
    print(f"Multiply: {multiply(10, 5)}")
    print(f"Divide: {divide(10, 5)}")
```

# OUTPUT:

```
Add: 15
Subtract: 5
Multiply: 50
Divide: 2.0
```