G. Yeshashwini

2303A52023

Batch - 38

Task Description-1 (Classes – Data Validation)

• Prompt AI to generate a Student class with attributes: name,

roll_no, and marks. Add a method is_pass() that returns whether

the student has passed (marks $\geq$ 40).

Expected Output-1

• Python class with constructor and validation logic

• Clear method implementation returning pass/fail status

• Students analyze correctness and clarity of class design

PROMPT:

Generate a Student class with attributes: name, roll_no, and marks. Add a method is_pass() that returns whether the student has passed (marks $\geq$ 40)

CODE:

```python
#generate a Student class with attributes: name,roll_no, and marks. Add a
class Student:
    def __init__(self, name, roll_no, marks):
        self.name = name
        self.roll_no = roll_no
        self.marks = marks

    def is_pass(self):
        return self.marks >= 40
# Taking user input
name = input("Enter student's name: ")
roll_no = input("Enter student's roll number: ")
marks = float(input("Enter student's marks: "))
# Creating a Student object
student = Student(name, roll_no, marks)
# Checking if the student has passed
if student.is_pass():
    print(f"{student.name} has passed.")
else:
    print(f"{student.name} has failed.")
```

OUTPUT:

```
● Enter student's name: sonu
 Enter student's roll number: 20
 Enter student's marks: 45
 sonu has passed.
```

```
 Enter student's name: sanj
 Enter student's roll number: 19
 Enter student's marks: 39
 sanj has failed.
```

EXPLANATION:

This program creates a Student class to store a student's name, roll number, and marks. The constructor saves these details when the object is created. The is_pass() method checks if the marks are 40 or more to decide pass or fail. The program takes input from the user and shows the result.

Task Description-2 (Loops – Pattern Generation)

• Ask AI to generate a function that prints a right-angled triangle

star pattern using a for loop. Then regenerate the same pattern

using a while loop.

Expected Output-2

• Correct pattern output using both loop types

• Logical loop structure with proper condition

PROMPT:

generate a function that prints a right-angled triangle star pattern using a for loop

CODE:

```python
def print_triangle_pattern(n):
    for i in range(1, n + 1):
        print('*' * i)
# Example usage:
n = int(input("Enter the number of rows for the triangle pattern: "))
print_triangle_pattern(n)
```

OUTPUT:

```
Enter the number of rows for the triangle pattern: 5
*
**
***
****
*****
```

PROMPT:

generate a function that prints a right-angled triangle star pattern using a while loop

CODE:

```
#generate a function that prints a right-angled triangle star pattern using a whil
def print_triangle_pattern(rows):
    i = 1
    while i <= rows:
        j = 1
        while j <= i:
            print("*", end="")
            j += 1
        print()
        i += 1
# Example usage:
user_input = int(input("Enter the number of rows for the triangle pattern: "))
print_triangle_pattern(user_input)
```

OUTPUT:

```
Enter the number of rows for the triangle pattern: 5
*
**
***
****
*****
```

EXPLANATION:

These two programs print a right-angled triangle star pattern. The first program uses a for loop where the number of stars printed in each line increases with every loop iteration. The second program uses while loops to achieve the same result by controlling the number of rows and stars using conditions. Both programs produce the same output but use different looping methods.

Task Description-3 (Conditional Statements – Number Analysis)

• Ask AI to write a function that checks whether a given number

is positive, negative, or zero using if-elif-else. Test the function

with multiple inputs.

Expected Output-3

• Function correctly classifies numbers

• Proper handling of all conditions

• Students analyze decision logic

PROMPT:

Write a function that checks whether a given number is positive, negative, or zero using if-elif-else

CODE:

```python
def check_number():
    try:
        num = float(input("Enter a number: "))
        if num > 0:
            print("The number is positive.")
        elif num < 0:
            print("The number is negative.")
        else:
            print("The number is zero.")
    except ValueError:
        print("Invalid input. Please enter a numeric value.")
# Call the function
check_number()
```

OUTPUT:

```
Enter a number: 12
The number is positive.
```

```
Enter a number: -12
The number is negative.
```

```
Enter a number: 0
The number is zero.
```

EXPLANATION:

This program asks the user to enter a number and checks whether it is positive, negative, or zero. It prints the result using if-elif-else. If the user enters something that is not a number, it shows an error message.

Task Description-4 (Nested Conditionals)

• Generate a function check_discount(age, is_member) that

determines discount eligibility:

• Age ≥ 60 → Senior discount

• Member → Additional discount

Use nested if statements.

Expected Output-4

• Python code using nested conditionals

• Clear explanation of decision flow

PROMPT:

Generate a function check_discount(age, is_member) that determines discount eligibility:

• Age ≥ 60 → Senior discount

 • Member → Additional discount

Use nested if statements

CODE

```python
def check_discount(age, is_member):
    if age >= 60:
        if is_member:
            return "Eligible for senior and member discounts."
        else:
            return "Eligible for senior discount only."
    else:
        if is_member:
            return "Eligible for member discount only."
        else:
            return "Not eligible for any discounts."
# Example usage:
age = int(input("Enter your age: "))
is_member_input = input("Are you a member? (yes/no): ").strip().lower()
is_member = is_member_input == 'yes'
result = check_discount(age, is_member)
print(result)
```

OUTPUT:

```
Enter your age: 40
Are you a member? (yes/no): YES
Eligible for member discount only.
```

EXPLANATION:

This program checks discounts using age and membership. If the person is 60 or older, they get a senior discount. If they are also a member, they get both discounts. If they are under 60, only members get a discount. If none of these apply, no discount is given.

Task Description-5 (Class – Mathematical Opera)

• Ask AI to create a Circle class with methods to calculate area ()

and circumference () given the radius.

Expected Output-5

• Correct mathematical computation

• Well-structured class with methods

• Code explanation provided

PROMPT:

create a Circle class with methods to calculate area () and circumference () given the radius

CODE:

```python
import math
class Circle:
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        return math.pi * (self.radius ** 2)

    def circumference(self):
        return 2 * math.pi * self.radius
# Get user input for radius
radius = float(input("Enter the radius of the circle: "))
circle = Circle(radius)
print(f"Area of the circle: {circle.area()}")
print(f"Circumference of the circle: {circle.circumference()}")
```

OUTPUT:

```
Enter the radius of the circle: 12
Area of the circle: 452.3893421169302
Area of the circle: 452.3893421169302
Circumference of the circle: 75.39822368615503
```

EXPLANATION:

This program creates a Circle class that stores the radius of a circle. It uses one method to find the area and another method to find the circumference. The radius is taken from the user, and the results are printed.