ASSIGNMENT-6

Roll no:2303A52074
Batch-37

Task Description-1 (Classes – Data Validation)

Expected Output
A Python class named Student .
It checks marks and tells whether the student has passed or failed.

Prompt
Create a class called Student with attributes name, roll_no, and marks. Add a method is_pass() that returns whether the student has passed (marks ≥ 40).give user defined values for name, roll_no, and marks while creating an object of the class.

Code

```python
#Create a class called Student with attributes name, roll_no, and marks. Add a method is_pass() that returns whether the student has passed
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
# User-defined values
name = input("Enter student's name: ")
roll_no = input("Enter student's roll number: ")
marks = float(input("Enter student's marks: "))
# Creating an object of the Student class
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 not passed.")

```

Output

```
Enter student's name: ram
Enter student's roll number: 2074
Enter student's marks: 88
ram has passed.
```

Explanation
Student details are stored in Student class.
It defines the name, roll number, and marks with the help of a constructor.
The marks value is verified, to make sure that it is valid.
The is_pass method gives the answer as to whether the student has passed or not**.**

Task Description-2 (Loops – Pattern Generation)

Expected Output
A right-angled triangle star pattern is printed.
The same pattern is generated using both for loop and while loop.
The loop logic is correct with proper conditions.

Prompt
Generate a Python function that prints a right-angled triangle star pattern using a for loop.Then generate the same pattern using a while loop.

Code

```
#Generate a Python function that prints a right-angled triangle star pattern using a for loop.Then generate the same pattern using a while
def print_triangle_for(n):
    for i in range(1, n + 1):
        print('*' * i)

def print_triangle_while(n):
    i = 1
    while i <= n:
        print('*' * i)
        i += 1
# Number of rows for the triangle
n = int(input("Enter the number of rows for the triangle: "))
print("Right-angled triangle pattern using for loop:")
print_triangle_for(n)
print("Right-angled triangle pattern using while loop:")
print_triangle_while(n)

```

Output

```
Enter the number of rows for the triangle: 5
Right-angled triangle pattern using for loop:
*
**
***
****
*****
Right-angled triangle pattern using while loop:
*
**
***
****
*****
```

Explanation
The program will be used to display a right-angled triangle with the help of stars.To start with, the pattern is printed with the help of a for loop.The process is repeated and then a white loop is used to print the same pattern.

Task Description-3 (Conditional Statements – Number Analysis)

Expected Output
A Python function that correctly identifies whether a number is positive, negative, or zero.
All conditions are handled properly using if elif else.

Prompt
Write a Python function that checks whether a given number is positive, negative, or zero using if,elif,else.Test the function with user defined multiple inputs.

Code

```python
#Write a Python function that checks whether a given number is positive, negative, or zero using if-elif-else.Test the function with user d
def check_number(num):
    if num > 0:
        return "Positive"
    elif num < 0:
        return "Negative"
    else:
        return "Zero"
# Testing the function with user-defined inputs
numbers = input("Enter numbers separated by spaces: ").split()
for number in numbers:
    num = float(number)
    result = check_number(num)
    print(f"{num} is {result}.")


```

Output

```
Enter numbers separated by spaces: 1 -1 2 3 -9 10
1.0 is Positive.
-1.0 is Negative.
2.0 is Positive.
3.0 is Positive.
-9.0 is Negative.
10.0 is Positive.
```

Explanation

The number is checked by the functionalities through conditional statements.
The number is printed in the form of positive, negative or zero depending on the value.
All the possible cases are managed in a clear way with the use of if-elif-else**.**

Task Description-4 (Nested Conditionals)

Expected Output

A Python function that uses nested if statements to check discount eligibility.
Senior discount is applied when age is 60 or above.

Prompt

Write a python function check_discount age, is_member, using nested if statements.If age is 60 or above, apply a senior discount.If the person is also a member, apply an additional discount.

Code

```python
#Write a python function check_discount age, is_member, using nested if statements.If age is 60 or above, apply a senior discount.If the pe
def check_discount(age, is_member):
    discount = 0
    if age >= 60:
        discount += 10  # Senior discount
        if is_member:
            discount += 5  # Additional member discount
    return discount
# Testing the function with user-defined inputs
age = int(input("Enter age: "))
is_member_input = input("Is the person a member? (yes/no): ").strip().lower()
is_member = is_member_input == 'yes'
total_discount = check_discount(age, is_member)
print(f"Total discount applicable: {total_discount}%")


```

Output

```
Enter age: 62
Is the person a member? (yes/no): no
Total discount applicable: 10%
```

Explanation

The function first checks if the age is 60 or above to apply a senior discount.
If this condition is true, it then checks whether the person is a member for an extra discount.
If the person is not a senior, only the membership condition is checked.
Nested if statements are used to check the discount logic clearly

Task Description-5 (Class – Mathematical Opera)

Expected Output

A Python CIRLCE class with a radius attribute.
Methods to correctly calculate the area and circumference of a circle

Prompt

Create a class Circle with methods to calculate area and circumference using the given radius.

Code

```
#Create a class Circle with methods to calculate area and circumference using the given radius.
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
# User-defined radius
radius = float(input("Enter the radius of the circle: "))
# Creating an object of the Circle class
circle = Circle(radius)
# Calculating area and circumference
area = circle.area()
circumference = circle.circumference()
print(f"Area of the circle: {area}")
print(f"Circumference of the circle: {circumference}")

```

Output

```
Enter the radius of the circle: 2
Area of the circle: 12.566370614359172
Circumference of the circle: 12.566370614359172
```

Explanation

The Circle class is used to represent a circle using its radius.It has one method to find the area and another to find the circumference.Both methods use standard math formulas to get correct results