**DESIGN PATTERN**

**Step 1-**

Given two triangles (one scalene and another isosceles) calculate the area and perimeter of both .Note that all three sides of a scalene triangle are available and only the base and one side is mentioned.(Solve using constructor overloading).

**Disadvantages**- Consider another triangle that is right angled here we cannot use constructor overloading since the no. of parameters in scalene and isosceles are same i. e 2 but the method of initializing the third side isn’t the same.

**Step 2-**

Write a code to calculate area and perimeter of all three types of triangles. (Hint: use super class method)

**Disadvantages of Super class method-**

* Increases the no. of classes ultimately resulting in a complicated project
* Inefficient for large projects
* The super class being a class should have attribute and behavior but that isn’t the case we just initialize the attributes, it doesn’t have any functions.

**Step 3-**

The same problem can be solved using menu driven approach using a variable (‘type’ describing the type of triangle) for calculating the area and perimeter of the three types of triangles.

**Disadvantages of menu driven approach-**

* The code becomes very monotonous and lengthy
* Repetition of code
* It is confusing as to what number refers to which type when menu is large

**Step 4-**

Solve this problem using the concept of static function.

**Solution-**

By declaring a **function** member as **static**, you make it independent of any particular object of the class. A **static** member **function** can be called even if no objects of the class exist and the **static functions** are accessed using only the class name and the scope resolution operator ::

* This type of design pattern is called **Factory Design Pattern**
* Some important points about Factory Design Pattern method are:

1. We can keep an independent factory class or we can keep a static method that returns the initialized object.

* **Factory Design Pattern Advantages**

1. It gives the user the privilege to create an object without having to bother about the internal details.
2. Allows lose coupling( i. e It supports the modification in the internal system without changing the API )

* Real Time Example where Factory Design Pattern can be used-

Consider an Airplane Ticket Reservation System where there are two types of tickets available Business class and economy. Following are the other parameters on the ticket

1. Meal(y/n)
2. Insurance(y/n)
3. Checking baggage(y/n)

Given that if the ticket is Business class all of the above parameters should be initialized as ‘yes’.

For more details visit-

<https://en.wikipedia.org/wiki/Factory_method_pattern>