Task 1: Computation class

1. Create a **Coputation class** with a default constructor (without parameters) allowing to perform various calculations on integers numbers.
2. Create a method called **Factorial()** which allows to calculate the factorial of an integer. Test the method by instantiating the class.
3. Create a method called **Sum()** allowing to calculate the sum of the first n integers **1 + 2 + 3 + .. + n.** Test this method.
4. Create a method called **testPrim()** in the Calculation class to test the primality of a given integer. Test this method.
5. Create a method called **testPrims()** allowing to test if two numbers are prime between them.
6. Create a **tableMult()** method which creates and displays the multiplication table of a given integer. Then create an **allTablesMult()** method to display all the integer multiplication tables 1, 2, 3, ..., 9.
7. Create a static **listDiv()** method that gets all the divisors of a given integer on new list called **Ldiv**. Create another **listDivPrim()** method that gets all the prime divisors of a given integer.

Task 2 : Shape Hierarchy (Inheritance & Polymorphism):

- Create an abstract base class `Shape` with an abstract method `area()`.
- Define subclasses `Rectangle`, `Circle`, and `Triangle` that inherit from `Shape`.
- Implement the `area` method in each subclass based on their specific formulas.
- Create a function `calculate_total_area(shapes)` that takes a list of `Shape` objects and calculates the total area by iterating through them and calling the appropriate `area` method (polymorphism).
- Test the function with different shapes and verify the correct calculation.