Exercise 01:

Try following code. What is the outcome? Why?

Class 01:

final class Student {

final int marks = 100;

final void display();

}

**Issues with Class 01:**

**final class Student: The class Student is declared as final. A final class cannot be subclassed, which means no other class can inherit from it.**

**final int marks = 100;: The variable marks is declared as final, which means it cannot be modified once assigned. However, there is no value assigned to it directly or through a constructor, so it will result in a compilation error.**

**final void display();: The method display() is declared as final, indicating that it cannot be overridden by subclasses. However, there is no method body or implementation for this method, resulting in a compilation error.**

Class 02:

class Undergraduate extends Student{}

**Issues with Class 02:**

**extends Student: The class Undergraduate is attempting to extend the Student class, but as mentioned earlier, the Student class is declared as final, and no class can extend it. This will cause a compilation error.**

**In summary, the code will not compile due to the following reasons:**

* **Class 01 is declared as final, preventing it from being subclassed (extended).**
* **Class 01 contains a final variable and a final method without appropriate implementations.**
* **Class 02 is attempting to extend the final class Student, which is not allowed.**

Exercise 02:

Develop a code base for the following scenario. Shape class contains an abstract method called “calculateArea” and non-abstract method called “display”. Try to pass required values at the instantiation. Recall what we have done at the lecture…

