Design an object model for a given domain. Use (demonstrate the ability to work with):

- Classes (abstract classes – if possible)

- Interfaces

- Inheritance,

- Polymorphism,

- Encapsulation

Each class methods and variables must have a sense and be informative. It is good to think about exactly what classes are needed in the solution. Inheritance should be used only when it makes sense. Classes must be properly laid out in packages. Working with a user’s console menu should be minimal (only necessary data input, display only what is asked in the problem). The task is a domain area where you have to allocate the necessary hierarchy of classes and to implement it with the features of OOP (using inheritance, if necessary or realizing interfaces). Each class must have fields and methods that you consider necessary. The program should create objects of different classes in the selected domain, combine them in any set of objects (use the collection). As a rule, the task requires to perform some action on the objects in a collection of objects based on specified criteria.

**Tasks :**

***Create the below project using Maven.***

1. New Year's gift. Identify the hierarchy of chocolates and other sweets. Create multiple objects of sweets. Collect children's gift to define total weight. Sort the chocolates in a gift by one of the options. Find candies in the gift corresponding to a predetermined range of options.