Lab1 : Inheritance and Polymorphism

Using an inheritance hierarchy, design a Java program to model items at a library (books, journal articles, videos and CDs.) Have an abstract superclass called Item and include common information that the library must have for every item (such as unique identification number, title, and number of copies). No actual objects of type Item will be created - each actual item will be an object of a (non-abstract) subclass. Place item-type-specific behavior in subclasses (such as a video's year of release, a CD's musical genre, or a book's author).
More in detail:

1. Implement an abstract superclass called Item and define all common operations on this class (constructors, getters, setters, print, checkIn, checkOut, addItem, etc). Have private data for: identification number, title, and number of copies.

2. Implement an abstract subclass of Item named WrittenItem and define all common operations on this class. Added private data for author.

3. Implement 2 subclasses of WrittenItem: Book and JournalPaper.

   3.1. Class Book: no new private data. When needed, override/overload methods from the superclass.
   3.2. Class JournalPaper: added private data for year published. When needed, override/overload methods from the superclass.

4. Implement another abstract subclass of Item named MediaItem and define all common operations on this class. Added private data for runtime (integer).

5. Implement 2 subclasses of MediaItem: Video and CD.

   5.1. Class Video: added private data for director, genre and year released. When needed, override/overload methods from the superclass.
   5.2. Class CD: added private data for artist and genre. When needed, override/overload methods from the superclass.

Write the definitions of these classes and a client program (your choice!) showing them in use.

Lab 2: Flow control and Exception Handling

Exercise 1: Write a java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On entering the choice, an appropriate message with “stop” or “ready” or “go” should appear in the console .Initially there is no message shown.

Exercise 2: The Fibonacci sequence is defined by the following rule. The first 2 values in the sequence are 1, 1. Every subsequent value is the sum of the 2 values preceding it. Write a Java program that uses both recursive and nonrecursive functions to print the nth value of the Fibonacci sequence?

Exercise 3: Write a Java program that prompts the user for an integer and then prints out all the prime numbers up to that Integer?

Exercise 4: Write a Java Program to validate the full name of an employee. Create and throw a user defined exception if firstName and lastName is blank.

Exercise 5: Validate the age of a person and display proper message by using user defined exception. Age of a person should be above 15.

Exercise 6: Create an Exception class named as “EmployeeException”(User defined Exception) in a package named as “com.cg.eis.exception” and throw an exception if salary of an employee is below than 3000. Use Exception Handling mechanism to handle exception properly.