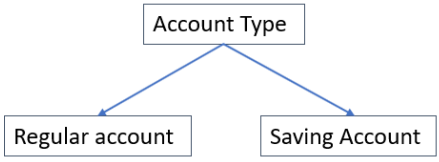
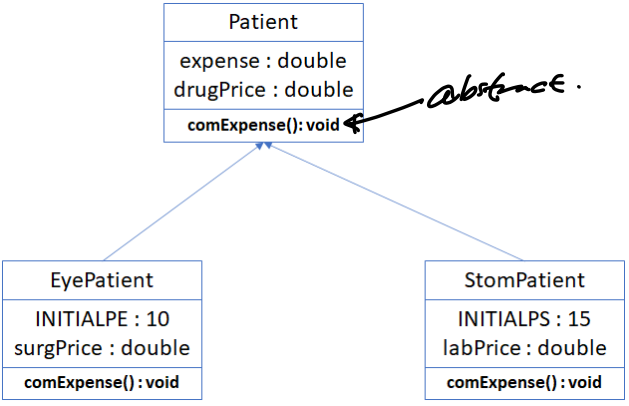
Object Oriented Programming with Java
Lab Practice:6

1. Consider the following class hierarchy.



- Declare an abstract class called Account which has the attributes Account Id, Name, Address, and balance and an abstract method called calculateInterest.
- Implement the abstract method in the class Savings Account. Use any simple interest formula to calculate the interest. (You can assume that you will get time/number of years as input to the method). You can leave the subclass Checking Account as abstract.
- Finally, compile your program.

2. Consider the following class hierarchy:



Write the code that implements these classes and their methods in Java, each class should contain the following:
(a) Default constructor.
(b) A constructor for an object with its all attributes.
(c) computeExpenses () in Patient class is simply an abstract method that needs to be

overridden in the subclasses.
(d) computeExpenses() in the subclasses overrides the inherited one as follows:

- computeExpenses() for the EyePatient stores the total expenses in the expense data member as follows:
-Expense = price of the Surgery+initial Payment +price of drugs
-computeExpenses () for the StomPatient stores the total expenses in the expense data member as follows:
-Expenses = initial Payment +price of drugs+ Lab.

(e) Create a testing class that declares a 1-D array called mClinic then enables 100 patient's objects to be inserted in mClinic whether they are of Eye-Patient type or of Stomach-Patient type.

* "equals (Object obj) method in java.lang
→ Simply Compare the references of obj and this with "==" not Content.

Call by Value

Call by References.

String a = "Hello World"
String b = a
String c = new String ("Hello World")

a == b ○ a == c ×
a.equals(b) ○ a.equals(c) ○