Object Oriented Programming with JAVA

Lab -4 StringBuffer, abstract class and final keyword

Core:

1 Create a StringBuffer object and illustrate as below:
- Refer to document via " javap –s java.lang.StringBuffer"
- Know length, capacity of object and content using toString
- Insert, append character
- Replacing, substring , ==, equals and compareTo

2 Define an abstract base class Shape that includes protected data members for the (*x*, *y*) position of a shape, a public method to move a shape, and a public abstract method show() to output a shape. Derive subclasses for lines, circles, and rectangles. You can represent a line as two points, a circle as a center and a radius, and a rectangle as two points on diagonally opposite corners. Implement the toString() method for each class. Test the classes by selecting ten random objects of the derived classes, and then invoking the show() method for each.

3 Write a program to demonstrate the all three uses of final keyword.

Plus:

1 Write a java program containing following classes with mentioned features.
- Person Class:
  - An instance variable full name of type string
  - A constructor with parameter to initialize full name
  - An abstract method getDescription with return type String
  - A concrete method getName() which returns full name of that person
- Employee Class:
  - An instance variable salary
  - A constructor with two parameter full name and salary
  - A method getSalary() which returns salary of an employee
  - An implementation of getDetail() method returns description of that person(i.e. employee) with salary value
- Student Class:
  - An instance variable branch
  - A constructor with two parameters full name and branch.
  - A method getBranch() which returns branch in which that student is studying
  - An implementation of getDetail() method returns description of that person(i.e. employee) with branch name.

- DemoAbstract class:
  - I n main method, create two variable person1 and person2 of Person type.
  - Initialize person1 with an object of Employee with necessary values
  - Initialize person2 with an object of Student with necessary values
  - Print description of each persons
- Sample output :
  - Manoj Pandey, an employee with the salary of Rs 50000
  - Rohini Dave, a student studying in Computer Science

2 Write a java program with classes Person and Employee as in above problem. Also write classes Manager and DemoFinal as per the following description

- Manager class:
  - A String instance variable type which stores value of manager type(i.e. HR, General, Finanace…).
  - A constructor with three parameters full name, salary and type.
  - An instance method increaseSalary() which takes an amount by which manager's salary will be incremented as a parameter and increased amount set as a new value to variable salary
  - An overridden method getDetail() which also prints the manager type of that manager.
- DemoFinal Class:
  - In main method, create an instance manager1 of type Manager with suitable initial values
  - Print Description of manager1
  - Call increaseSalary() on manager1 by passing some appropriate value
  - Print Description of manager1 again
- Execution of program in different cases:
  - Case A: Make an Employee class Final
  - Case B: Make a method getDescription() in employee class as a final method
  - Case C: Make an instance variable salary in class Employee.