**Practical List-**

1. Create a class “Student” having following instance variables and methods.

**Instance variables:** ID, Name, Branch and university

**Method:** setDetails() and showDetails().

The setDetails() method sets the values of ID, Name, Branch and University.

And showDetails() method shows the value of each field.

2. Write a Java Program to demonstrate the working of a banking-system

**Instance variables:** name, account\_no, amount

**Instance methods:**  deposit(), withdraw(), checkBalance(), insert() and display().

Here we can deposit and withdraw amount from our account using deposit() and withdraw() methods respectively.

The insert() method is to initialize state and display() method is to display state values.

3. Write a program to sum two numbers. Here inputs are provided through command line argument.

4. Create class Employee with following attributes and methods

ID, name, department and salary.

The setDetails() method sets the values of ID, name, department and salary.

And showDetails() method shows the value of each field.

**Note:** (i) Values must be entered through Scanner class.

(ii) Use proper constructor

(iii) Use “this” reference variable to avoid ambiguity.

5. Re-write program 1 with better memory management approach.

**Note:** use of static keyword

6. Apply following functions on the String "Java".

(i) Try to concat "Welcome" and write down your observation.

(ii) Find character at index 1

(iii) Find index of first 'a'.

(iv) Find index of second 'a'

(v) Compare "Java" to "JAVA"

(vi) Compare "Java" to "JAVA" ignoring the case

(vii) Find the index of first 'a' from last

7. Apply following functions on StringBuffer object "HELLO"

(i) Append "Java"

(ii) Insert "Java" at index 1

(iii) Replace with "Java" with characters between index 1 to 2

(iv) Delete characters between index 1 and 2

(v) Reverse the string "HELLO"

8. Create a class “Student” having following instance variables and methods.

Instance variables: ID, Name, Branch, city and university

While creating constructors with one, two, three, four and five arguments reuse the constructors by **construction chaining**

9. Create two dimensional integer array and insert, search and traverse this array.

**Note:** Use Scanner class to insert data.

10. Create a jagged array having three rows. Where 1st row contains 3 columns, 2nd row contains 4 columns and 3rd row contains 2 columns. Insert and traverse it.

11. Create a class “Shape” having area() method to calculate area. Overload the area() method for shapes like triangle, rectangle and circle.

12. Create a class “Bank” having method getRateOfInterest(). Create child classes as HDFC, SBI and PNB and override getRateOfInterest() and return interest rates as 4.0, 4.5 and 5% correspondingly.

**Use concept of Upcasting to implement this scenario.**