Practical List

S.No.	Practical Name	Page No.
1	Swapping of two numbers without accepting numbers from the user.	4-5
2	Find out the largest of two numbers using command line arguments.	6-7
3	Print class of students according to given range using switch case.	8-9
4	Find GCD and LCM of two positive numbers using a while loop.	10-11
5	Display table of given number using for loop.	12-13
6	Java Program to check whether input character is vowel or consonant	14-16
7	Print the upper triangle using a nested for loop. * * * * * * * * * *	17-19
8	Java Program to calculate simple interest and compound interest.	20-21
9	Java program to find occurrence of a character in a String.	22-24
10	Write a class Circle with methods getdata() and area(). Create an object of that class, find the area and display the result.	25-27
11	Write a program to implement single inheritance using super keywords.	28-29
12	Write a program to add two numbers using single inheritance such that the base class method must accept 2 numbers from the user and the derived class method must add these numbers and display the sum.	30-32
13	Write a program to implement multiple inheritance.	33-35
14	Write a program to print 1A2B3C4D5E6F7G8H9I10J using thread.	36-37

Aim: Swapping of two numbers without accepting numbers from the user.

Source Code:

```
import java.util.*;
class Swap_With {
public static void main(String[] args) {
int x, y, t;// x and y are to swap
Scanner sc = new Scanner(System.in);
System.out.println("Enter the value of X and Y");
x = sc.nextInt();
y = sc.nextInt();
System.out.println("before swapping numbers: "+x +"  "+ y);
/*swapping */
t = x;
x = y;
y = t;
System.out.println("After swapping: "+x +"   " + y);
System.out.println( );
}
}
```

Sample Input:
Enter the value of X and Y
2
3

Expected Output:
before swapping numbers: 2 3
After swapping: 3 2

Aim: Find out the largest of two numbers using command line arguments.

Source Code:

```
import java.util.Scanner;
public class LargestNumberExample1
{
public static void main(String[] args)
{
int a, b, c, largest, temp;
//object of the Scanner class
Scanner sc = new Scanner(System.in);
//reading input from the user
System.out.println("Enter the first number:");
a = sc.nextInt();
System.out.println("Enter the second number:");
b = sc.nextInt();
System.out.println("Enter the third number:");
c = sc.nextInt();
//comparing a and b and storing the largest number in a temp variable
temp=a>b?a:b;
//comparing the temp variable with c and storing the result in the variable
largest=c>temp?c:temp;
//prints the largest number
System.out.println("The largest number is: "+largest);
}
}
```

Sample Input:
Enter the value of X and Y
2
3

Expected Output:
before swapping numbers: 2 3
After swapping: 3 2

Aim: Print class of students according to given range using switch case.

Source Code:

```
public class GradeCalculator {

  public static void main(String[] args) {
    // score (0-100)
    int score = 79;
    String grade = null;

    switch(score/10) {
      // for >= 90
      case 10:
      case 9:
        grade = "A";
        break;
      // for >= 80 and <90
      case 8:
        grade = "B";
        break;
      // for >= 70 and <80
      case 7:
        grade = "C";
        break;
      // for >= 60 and <70
      case 6:
        grade = "D";
        break;
      // for >= 50 and <60
      case 5:
        grade = "E";
        break;
      // for < 50
      default:
        grade = "F";
        break;
    }

    // display result
    System.out.println("Grade = " + grade);
  }

}
```

Sample Input: Grade : C

Expected Output: Grade = C

Aim: Find GCD and LCM of two positive numbers using a while loop.

Source Code:

```
public class FindGCDExample1
{
public static void main(String[] args)
{
//x and y are the numbers to find the GCF
int x = 12, y = 8, gcd = 1;
//running loop form 1 to the smallest of both numbers
for(int i = 1; i <= x && i <= y; i++)
{
//returns true if both conditions are satisfied
if(x%i==0 && y%i==0)
//storing the variable i in the variable gcd
gcd = i;
}
//prints the gcd
System.out.printf("GCD of %d and %d is: %d", x, y, gcd);
}
}
```

Sample Input: GCD of 12 and 8 is: 4

Expected Output: GCD of 12 and 8 is: 4

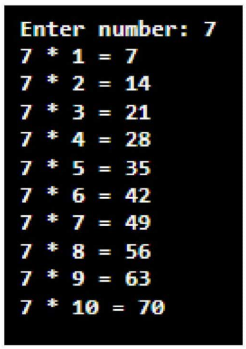
Aim: Display table of given number using for loop

Source Code:

```
import java.util.Scanner;
public class TableExample
{
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
System.out.print("Enter number: ");
//reading a number whose table is to be print
int num=sc.nextInt();
//loop start execution form and execute until the condition i<=10 becomes false
for(int i=1; i <= 10; i++)
{
//prints table of the entered number
System.out.println(num+" * "+i+" = "+num*i);
}
}
}
```

Sample Input: Enter Number:7

Expected Output:



Aim: Java Program to check whether input character is vowel or consonant

Source Code:

```
import java.io.*;

public class geek {

    // Function to find whether an input

    // character is vowel or not

    static void Vowel_Or_Consonant(char y)

    {

        if (y == 'a' || y == 'e' || y == 'i' || y == 'o'

            || y == 'u')

            System.out.println("It is a Vowel.");

        else

            System.out.println("It is a Consonant.");

    }


    // The Driver code

    static public void main(String[] args)

    {

        Vowel_Or_Consonant('b');

        Vowel_Or_Consonant('u');
```

}

}

Input : char = 'e'
Output : Vowel

Aim: Print the upper triangle using a nested for loop

Source Code:

```
public class UpperTriangular
{
   public static void main(String[] args) {
      int rows, cols;

      //Initialize matrix a
      int a[][] = {
                    {1, 2, 3},
                    {8, 6, 4},
                    {4, 5, 6}
                  };

       //Calculates number of rows and columns present in given matrix
       rows = a.length;
      cols = a[0].length;

      if(rows != cols){
         System.out.println("Matrix should be a square matrix");
      }
      else {
         //Performs required operation to convert given matrix into upper triangular matrix
         System.out.println("Upper triangular matrix: ");
         for(int i = 0; i < rows; i++){
            for(int j = 0; j < cols; j++){
              if(i > j)
                System.out.print("0 ");
              else
                System.out.print(a[i][j] + " ");
            }
            System.out.println();
         }
      }
   }
}
```

Sample Input:
matrix a[][] ={{1,2,3},{8, 6, 4}, {4, 5, 6}}

Expected Output:
Upper triangular matrix:
1 2 3
0 6 4
0 0 0

Aim: Java Program to calculate simple interest and compound interest

Source Code:

```
public class Main {
  public static void main(String[] args) {

    System.out.print("Principal = 9000: ");
    double principal = 9000;

    System.out.print("rate of interest = 10 : ");
    double rate = 10;

    System.out.print("number of years = 5 : ");
    int years = 5;

    double simpleInterest = (principal * rate * years) / 100;
    System.out.println("Simple Interest: " + simpleInterest);
  }
}
```

Sample Input:
Principal = 9000: rate of interest = 10 : number of years = 5 :

Expected Output:

Principal = 50000.0
Annual Rate of Interest = 20.0
Time (years) = 5.0
Simple Interest: 50000.0
Compound Interest: 74415.99999999997

Aim : Java program to find occurrence of a character in a String.

Source Code:

```
public class CountOccuranceOfChar1
{
static final int MAX_CHAR = 256;
static void getOccuringChar(String str)
{
//creating an array of size 256 (ASCII_SIZE)
int count[] = new int[MAX_CHAR];
//finds the length of the string
int len = str.length();
//initialize count array index
for (int i = 0; i < len; i++)
count[str.charAt(i)]++;
//create an array of given String size
char ch[] = new char[str.length()];
for (int i = 0; i < len; i++)
{
ch[i] = str.charAt(i);
int find = 0;
for (int j = 0; j <= i; j++)
{
//if any matches found
if (str.charAt(i) == ch[j])
find++;
}
if (find == 1)
//prints occurrence of the character
System.out.println("The occurrence of "+ str.charAt(i)+ " is: " + count[str.charAt(i)]);
}
}
//driver Code
public static void main(String args[])
{
String str = "Pneumonoultramicroscopicsilicovolcanoconiosis"; //lung disease
//function calling
getOccuringChar(str);
}
}
```

Sample Input:

String str = "Pneumonoultramicroscopicsilicovolcanoconiosis";

Expected Output:

The occurrence of P is: 1
The occurrence of n is: 4
The occurrence of e is: 1
The occurrence of u is: 2
The occurrence of m is: 2
The occurrence of o is: 9
The occurrence of l is: 3

The occurrence of t is: 1
The occurrence of r is: 2
The occurrence of a is: 2
The occurrence of i is: 6
The occurrence of c is: 6
The occurrence of s is: 4
The occurrence of p is: 1
The occurrence of v is: 1

Aim : Write a class Circle with methods getdata() and area(). Create an object of that class, find the area and display the result.

Source Code:

```
// Java program to find the area of
// the circle using Method Overloading

import java.io.*;

class Circle {

        static final double PI = Math.PI;

        // Overloaded Area() function to
```

```
        // calculate the area of the circle.
        // It takes one double parameter
        void Area(double r)
        {
                double A = PI * r * r;

                System.out.println("Area of the circle is :" + A);
        }

        // Overloaded Area() function to
        // calculate the area of the circle.
        // It takes one float parameter
        void Area(float r)
        {
                double A = PI * r * r;

                System.out.println("Area of the circle is :" + A);
        }
}

class GFG {

        // Driver code
        public static void main(String[] args)
        {

                // Creating object of Circle class
                Circle obj = new Circle();

                // Calling function
                obj.Area(5);
                obj.Area(2.5);
        }
}
```

Sample Input / Expected Output:

```
Area of the circle is :78.53981633974483
Area of the circle is :19.634954084936208
```

Mitte Mai Milla Denge

Aim: Write java a program to implement single inheritance using super keywords.

Source code:

```
class Animal{
String color="white";
}
class Dog extends Animal{
String color="black";
void printColor(){
System.out.println(color);//prints color of Dog class
System.out.println(super.color);//prints color of Animal class
}
}
class TestSuper1{
```

```
public static void main(String args[]){
Dog d=new Dog();
d.printColor();
}}
```

Sample Input: / Expected Output:

black
white

Aim : Write a program to add two numbers using single inheritance such that the base class method must accept 2 numbers from the user and the derived class method must add these numbers and display the sum.

Source Code:

```
class Calculation{

  int z;

  public void addition(int x, int y){

    z = x+y;

    System.out.println("The sum of the given numbers:"+z);

  }
```

```
  public void Substraction(int x,int y){
    z = x-y;
    System.out.println("The difference between the given numbers:"+z);
  }
 }
public class My_Calculation extends Calculation{
   public void multiplication(int x, int y){
    z = x*y;
    System.out.println("The product of the given numbers:"+z);
  }
  public static void main(String args[]){
    int a = 20, b = 10;
    My_Calculation demo = new My_Calculation();
    demo.addition(a, b);
    demo.Substraction(a, b);
    demo.multiplication(a, b);
  }
}
```

Sample Input:
a = 20, b = 10;

Expected Output:

The sum of the given numbers:30

The difference between the given numbers:10

The product of the given numbers:200

Aim: Write a program to implement multiple inheritance.

Source Code:

```
interface Character {
    void attack();
}

interface Weapon {
    void use();
}

class Warrior implements Character, Weapon {
    public void attack() {
        System.out.println("Warrior attacks with a sword.");
    }

    public void use() {
        System.out.println("Warrior uses a sword.");
    }
}
```

```
class Mage implements Character, Weapon {
    public void attack() {
        System.out.println("Mage attacks with a wand.");
    }

    public void use() {
        System.out.println("Mage uses a wand.");
    }
}

public class MultipleInheritance {
    public static void main(String[] args) {
        Warrior warrior = new Warrior();
            Mage mage = new Mage();

            warrior.attack(); // Output: Warrior attacks with a sword.
            warrior.use(); // Output: Warrior uses a sword.

            mage.attack(); // Output: Mage attacks with a wand.
            mage.use(); // Output: Mage uses a wand.
        }
    }
```

Sample Input / Expected Output:
Warrior attacks with a sword.
Warrior uses a sword.
Mage attacks with a wand.
Mage uses a wand.

Aim : Write a program to print 1A2B3C4D5E6F7G8H9I10J using thread.

Source Code:

```
class A extends Thread {
	public void run() {
		int i;
		for(i=1;i<=10;i++) {
			System.out.print(i);
			try {
				sleep(1000);
			} catch(Exception e){}
		}
	}
}
class B extends Thread {
	public void run() {
		int i;
		for(i=65;i<=74;i++) {
			System.out.print((char)i);
			try {
				sleep(1000);
			} catch(Exception e){}
		}
	}
}
```

```
class Multithread {
        public static void main(String args[]) {
                A t1=new A();
                B t2=new B ();
                t1.start();
                t2.start();
        }
}
```

Sample Input / Expected Output:
1A2B3C4D5E6F7G8H9I10J