

1. Write a Java program that reads a series of numbers from a file (input.txt). Determine the highest number in the series. Calculate the sum of natural numbers up to that highest number and write the result to another file (output.txt).

code:

```
import java.io.*;
import java.util.*;
public class main {
    public static void main (String [] args) throws Exception {
        Scanner sc = new Scanner (new File ("input.txt"));
        PrintWriter pw = new PrintWriter ("output.txt");
        String [] a = sc.nextLine ().split (" ");
        for (int i=0; i < length; i++) {
            int n = Integer.parseInt (a[i].trim ());
            pw.print (n * (n+1)/2);
            if (i < a.length - 1) pw.print (" ");
        }
        sc.close ();
        pw.close ();
    }
}
```

2. What are differences you have ever found between static and final fields and methods?

Feature	static field	final field
Belongs to	class	object/class
Value change	Allowed	Not Allowed
Memory	single copy	Depends
Example	static int x;	final int x=10;

Feature	static method	final method
OVERRIDING	Not overridden	Not overridden
Access	class name	Object name
Purpose	class-level logic	prevent overridden

Example:

```

class A{
    static int x=10;
    static void show(){
        System.out.println(x);
    }
    public static void main(String[] args){
        A.a=new A(); a.show();
    }
}
  
```

3. Write a java program to find all factorians within a given range.

code:

```
import java.util.Scanner;  
public class Factorian {  
    static int fact(int d) {  
        int f = 1;  
        for (int i = 1; i <= d; i++)  
            f = f * i;  
        return f;  
    }  
    static boolean isFactorian (int n) {  
        int sum = 0; temp = n;  
        while (temp > 0) {  
            sum = sum + fact(temp % 10);  
            temp /= 10;  
        }  
        return sum == n;  
    }  
    public static void main (String [] args) {  
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter the lower bound");  
        System.out.print ("Enter the upper bound");  
        System.out.print (Factorian numbers in the range);  
    }  
}
```

```

        for( int i = low; i<=high; i++ )
            if( isFactorian(i) )
                System.out.print( i + " " );
}
}

```

4. Distinguish the difference among class local and instance variables. What is significance of the keywords?

Ans:

Feature	Class Variable	Instance Variable	Local Variable
Keyword	Static	No Keyword	Declare inside method
Belongs to	one copy shared	object	method
Memory	Yes	seperated for each object	Creates when method runs
Default value	Class name	Yes	No
Access	program ends	Object	only side method
Lifetime	class	Object exists	Method execution

Example: class Test {
 static int a = 10; → class variable
 int b = 2; → instance variable
 void show() {
 int c = 30; → local variable
 }
}

. significance of this keyword

- Differentiates instance variables from parameter
- refers to current object
- class another constructor

5. Write a Java program that defines a method to calculates the sum of all elements in an integer arrays.

code: public class ArraySum {
 static int sumArray (int [] a) {
 int sum = 0;
 for (int i = 0; i < a.length; i++)
 sum = sum + a[i];
 return sum;
 }
}

```

public static void main(String []args){
    int []arr = {10, 20, 30, 40};
    int result = sumArray(arr);
    System.out.println("sum = " + result);
}
}

```

6. What is called Access modifier? Compare the accessibility of public, private and protected modifier. Describe different types of variables in Java with example

= Access modifier: Access modifier in Java is a keyword that defines the visibility of classes methods and variables.

Modifier	Accessible ie within class	Within package	Subclass	Everywhere
public	yes	yes	yes	yes
private	yes	NO	NO	NO
protected	yes	yes	yes	NO

Example:

```
public class Main {
    public static void main(String [] args) {
        new Test().show(5);
    }
}
```

```
(class Test) returning two states
public int a = 10; → instance variable
static int b = 20; → class/static variable
void show(int c) { → local variable
    System.out.println(a+b+c);
}
}
```

7. Write a Java program to find the smallest positive root of a quadratic equation of the form - $ax^2 + bx + c = 0$

using the quadratic formula, $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

Code:

```
import java.util.Scanner;
public class Smallest Root {
    public static void main (String [] args) {
```

```
scannen. sc = new Scanner(System.in);  
int a = sc.nextInt(), b = sc.nextInt();  
double D = b*b - a*a*c;  
if (D < 0)  
    System.out.println("No real roots");  
else {  
    double r1 = (-b + Math.sqrt(D)) / (2*a);  
    double r2 = (-b - Math.sqrt(D)) / (2*a);  
    if (r1 > 0 && r2 > 0)  
        System.out.out.println(Math.min(r1, r2));  
    else if (r1 > 0) System.out.println(r1);  
    else if (r2 > 0) System.out.println(r2);  
    else System.out.println("No possible  
roots.");  
}
```

9

IT-24030

8. Write a program that can determine the letter, whitespace and digit. How do we pass an array to a function? Write an example.

code:

```
import java.util.Scanner;  
public class main {  
    static void checkchar(char ch) {  
        System.out.println(Character.isLetter(ch) ? "Letter",  
                           Character.isDigit(ch) ? "Digit",  
                           Character.isWhitespace(ch) ? "Whitespace");  
    }  
    static void printArray(int []a) {  
        for (int i:a) System.out.println(i + " ");  
    }  
    static void printArray (int []a) {  
        for (int  
            Scanner sc = new Scanner(System.in),  
            checkchar (sc.next().charAt(0)),  
            printArray (new int [] {1, 2, 3, 4});  
        }  
    }  
}
```