

# WEEK 1

## EX 1.1

You are required to write a Java program to calculate the total salary of an employee based on their hourly wage, hours worked in a week, and the number of weeks they worked. The program should consider the following rules:

- If an employee works more than 40 hours in a week, they are paid 1.5 times their hourly wage for the overtime hours.
- If an employee works less than 20 hours in a week, they are penalized with a deduction of 10% of their weekly salary.
- The program should handle invalid inputs (e.g., negative values for hours or wages).

```
import java.util.*;

public class EmployeeSalaryCalculator{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter hourly wage: ");
        double hourlyWage = scanner.nextDouble();

        System.out.print("Enter hours worked per week: ");
        int hoursPerWeek = scanner.nextInt();

        System.out.print("Enter number of weeks worked: ");
        int weeksWorked = scanner.nextInt();

        if (hourlyWage <= 0 || hoursPerWeek < 0 || weeksWorked < 0) {
            System.out.println("Invalid input! Please enter positive values for wage, hours, and weeks.");
            return;
        }
        double totalSalary = 0;

        for (int i = 0; i < weeksWorked; i++) {
            double weeklySalary;

            if (hoursPerWeek > 40) {
                int overtimeHours = hoursPerWeek - 40;
```

```

        weeklySalary = (40 * hourlyWage) + (overtimeHours * hourlyWage * 1.5);
    }
    else if (hoursPerWeek < 20) {
        weeklySalary = hoursPerWeek * hourlyWage;
        weeklySalary -= weeklySalary * 0.1; // Deduct 10%
    }
    else {
        weeklySalary = hoursPerWeek * hourlyWage;
    }

    totalSalary += weeklySalary;
}

System.out.printf("Total salary is %.2f\n", totalSalary);
}
}

```

## OUTPUT-

```

PS D:\230701118\java(OOPS)> javac EmployeeSalaryCalculator.java
PS D:\230701118\java(OOPS)> java EmployeeSalaryCalculator
Enter hourly wage: 15.0
Enter hours worked per week: 45
Enter number of weeks worked: 4
Total salary is 2850.00

```

## EX 1.2

You are required to calculate the total cost of purchasing tickets for an event based on the ticket type and the number of tickets bought.

The program should consider the following rules:

- Regular Ticket: 50 each. If more than 10 tickets are bought, a discount of 10% is applied.
- VIP Ticket: 100 each. If more than 5 tickets are bought, a discount of 15% is applied.
- Premium Ticket: 150 each. If more than 3 tickets are bought, a discount of 20% is applied.

- If the total cost before any discount is less than 200, an additional service fee of 20 is applied.
- The program should handle invalid inputs (e.g., negative values for number of tickets, or invalid ticket types).

```
import java.util.*;
import java.lang.*;
public class Bill{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the ticket type:");
        String ticketType=sc.nextLine();
        System.out.println("Enter the number of tickets");
        int n=sc.nextInt();

        double costperTicket=0;
        double discount=0;
        if(ticketType.equalsIgnoreCase("Regular"))
        {
            costperTicket=50;
            if(n>10){
                discount=0.10;
            }
        }
        else if(ticketType.equalsIgnoreCase("Vip"))
        {
            costperTicket=100;
            if(n>5){
                discount=0.15;
            }
        }
        else if(ticketType.equalsIgnoreCase("Premium"))
        {
            costperTicket=150;
            if(n>3){
                discount=0.20;
            }
        }
    }
}
```

```

    }
    double totalcost=n*costperTicket;
    if(discount>0){
        totalcost-=totalcost*discount;
    }
    if(totalcost<200){
        totalcost+=20;
    }
    System.out.printf("Total cost is %.2f\n",totalcost);
}
}

```

## OUTPUT-

```

PS D:\230701118\java(OOPS)> javac Bill.java
PS D:\230701118\java(OOPS)> java Bill
Enter the ticket type:
Regular
Enter the number of tickets
12
Total cost is 540.00

```

## EX 1.3

Given a number N. The task is to find the largest and the smallest digit of the number.

```

import java.util.*;
class Large{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int s=9;
        int l=0;
        while(n>0){
            int digit=n%10;
            if(digit<s){
                s=digit;
            }
            if(digit>l){

```

```

        l=digit;
    }
    n=n/10;
}
System.out.println(s+" "+l);
}
}

```

## OUTPUT-

```

PS D:\230701118\java(OOPS)> javac Large.java
PS D:\230701118\java(OOPS)> java Large
2346
2 6
PS D:\230701118\java(OOPS)> java Large
956
5 9

```

## EX 1.4

This problem to understand the nested loop. Given N, a Positive integer, You are supposed to print the alternating 1's and 0's in triangle format.

```

import java. util.*;
public class Alternatetriangle{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        for(int i=1;i<=n;i++){
            int s=i%2;
            for(int j=1;j<=i;j++){
                System.out.print(s+" ");
                s=1-s;
            }
            System.out.println();
        }
    }
}

```

## OUTPUT-

```
PS D:\230701118\java(OOPS)> javac Alternatetriangle.java
PS D:\230701118\java(OOPS)> java Alternatetriangle

5
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

b)

Given N, a Positive integer, You are supposed to print in the below format.

```
import java. util.*;
public class PatternTriangle{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        for(int i=n;i>=1;i--){
            for(int j=1;j<=i;j++){
                System.out.print(j+" ");
            }
            System.out.println();
        }
    }
}
```

## OUTPUT-

```
PS D:\230701118\java(OOPS)> javac PatternTriangle.java
PS D:\230701118\java(OOPS)> java PatternTriangle

5
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

## EX 1.5

You are developing a scheduling application where users can check whether a given day is a weekday or a weekend. The application should prompt the user to enter a day of the week (e.g., "Monday", "Saturday"), and based on the input, the program should determine if the day is a weekday or a weekend.

```
import java.util.*;
public class Weekend{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        String s= sc.nextLine();
        switch(s){
            case "Monday":
            case "Tuesday":
            case "Wednesday":
            case "Thursday":
            case "Friday":
                System.out.println("It's a Weekday.");
                break;
            case "Saturday":
            case "Sunday":
                System.out.println("It's a Weekend.");
                break;
        }
    }
}
```

OUTPUT-

```

PS D:\230701118\java(OOPS)> javac Weekend.java
PS D:\230701118\java(OOPS)> java Weekend
Sunday
It's a Weekend.
PS D:\230701118\java(OOPS)> javac Weekend.java
PS D:\230701118\java(OOPS)> java Weekend
Monday
It's a Weekday.

```

## EX 1.6

Write a program to check whether a number is a Strong Number or not.

A strong number is a positive integer whose sum of the factorials of its digits equals the original number

Few examples of strong numbers are : 1,2,145 and 40585.

```

import java. util.*;
public class Strong{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int o=n;
        int sum=0;
        while(n>0){
            int rem=n%10;
            int f=1;
            for(int i=1;i<=rem;i++){
                f*=i;
            }
            sum+=f;
            n/=10;
        }
        if(sum==o){
            System.out.println("Strong Number.");
        }
        else{
            System.out.println("Not Strong Number.");
        }
    }
}

```



```
    }  
}
```

OUTPUT-

```
PS D:\230701118\java(OOPS)> javac Strong.java  
PS D:\230701118\java(OOPS)> java Strong  
145  
Strong Number.
```