WEEK ONE ASSIGNMENT:-

1. import java.util.Scanner;  
  
public class DataTypeFitter {  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        final long SHORT\_MIN = -32768;  
        final long SHORT\_MAX = 32767;  
        final long INT\_MIN = -2147483648L;  
        final long INT\_MAX = 2147483647L;  
        final long LONG\_MIN = -9223372036854775808L;  
        final long LONG\_MAX = 9223372036854775807L;  
  
               int numCases = scanner.nextInt();  
        scanner.nextLine();  
        for (int i = 0; i < numCases; i++) {  
             
            long number = scanner.nextLong();  
  
             
            System.out.println(number + " can be fitted in:");  
  
            if (number >= SHORT\_MIN && number <= SHORT\_MAX) {  
                System.out.println("\* short");  
            }  
            if (number >= INT\_MIN && number <= INT\_MAX) {  
                System.out.println("\* int");  
            }  
            if (number >= LONG\_MIN && number <= LONG\_MAX) {  
                System.out.println("\* long");  
            }  
  
                       if (!(number >= SHORT\_MIN && number <= LONG\_MAX)) {  
                System.out.println(number + " can't be fitted anywhere.");  
            }  
        }  
  
  
  }  
}

2.

import java.util.Scanner;  
import java.text.DecimalFormat;  
  
public class CentsToDollars {  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        DecimalFormat df = new DecimalFormat("0.00");         
        System.out.print("Enter amount in cents: ");  
        int cents = scanner.nextInt();  
  
  
        double dollars = cents / 100.0;  
  
         
        System.out.println("Amount in dollars: " + df.format(dollars));  
  
          
    }  
}

3.

public class ScoreDisplay {  
  
    public static void main(String[] args) {  
        double score1 = 456.89;  
        int intScore1 = (int) score1;  
        System.out.println("The score after type casting to int is " + intScore1);

        double score2 = 1234.56;  
        int intScore2 = (int) score2;  
        System.out.println("The score after type casting to int is " + intScore2);  
    }  
}

4.

public class PayrollSystem {  
  
    public static void main(String[] args) {  
        int initialSalary1 = 45000;  
        double percentageIncrease1 = 7.5;  
        double newSalary1 = calculateNewSalary(initialSalary1, percentageIncrease1);  
        System.out.println("The new salary after a " + percentageIncrease1 + "% increase is " + newSalary1);

        int initialSalary2 = 32000;  
        double percentageIncrease2 = 12.3;  
        double newSalary2 = calculateNewSalary(initialSalary2, percentageIncrease2);  
        System.out.println("The new salary after a " + percentageIncrease2 + "% increase is " + newSalary2);  
    }  
  
    public static double calculateNewSalary(int initialSalary, double percentageIncrease) {  
         
        double increaseAmount = initialSalary \* (percentageIncrease / 100.0);  
        double newSalary = initialSalary + increaseAmount;  
        return newSalary;  
    }  
}

1.

import java.util.Scanner;  
  
public class ReverseNumberChecker {  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
         
        System.out.print("Enter an integer: ");  
        int originalNumber = scanner.nextInt();  
         
        int reversedNumber = reverseNumber(originalNumber);  
         
        if (originalNumber == reversedNumber) {  
            System.out.println("The reversed number is " + reversedNumber + ". It is the same as the original.");  
        } else {  
            System.out.println("The reversed number is " + reversedNumber + ". It is not the same as the original.");  
        }  
         
        scanner.close();  
    }

    public static int reverseNumber(int number) {  
        int reversed = 0;  
        int original = number;  
         
        while (number != 0) {  
            int digit = number % 10;  
            reversed = reversed \* 10 + digit;  
            number /= 10;  
        }  
         
        return reversed;  
    }  
}

2.

import java.util.Scanner;  
  
public class DiamondPattern {  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
         
        System.out.print("Enter the number of rows for the top half of the diamond: ");  
        int n = scanner.nextInt();  
         
         for (int i = 1; i <= n; i++) {  
            for (int j = i; j < n; j++) {  
                System.out.print(" ");  
            }  
            for (int k = 1; k <= (2 \* i - 1); k++) {  
                System.out.print("\*");  
            }  
            System.out.println();  
        }  
         
        for (int i = n - 1; i >= 1; i--) {  
            for (int j = n; j > i; j--) {  
                System.out.print(" ");  
            }  
             for (int k = 1; k <= (2 \* i - 1); k++) {  
                System.out.print("\*");  
            }  
            System.out.println();  
        }  
         
        
    }  
}

3.

Import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
public class PascalHalfDiamond {  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
         
        System.out.print("Enter the number of rows for Pascal's Triangle: ");  
        int n = scanner.nextInt();  
         
          
        List<List<Integer>> pascalTriangle = generatePascalTriangle(n);  
         
        printPattern(pascalTriangle);  
         
        scanner.close();  
    }  
    public static List<List<Integer>> generatePascalTriangle(int n) {  
        List<List<Integer>> triangle = new ArrayList<>();  
         
        for (int i = 0; i < n; i++) {  
            List<Integer> row = new ArrayList<>();  
            row.add(1);   
            for (int j = 1; j < i; j++) {  
                 row.add(triangle.get(i - 1).get(j - 1) + triangle.get(i - 1).get(j));  
            }  
            if (i > 0) row.add(1);   
            triangle.add(row);  
        }  
         
        return triangle;  
    }  
  
      
    public static void printPattern(List<List<Integer>> pascalTriangle) {  
        int n = pascalTriangle.size();  
         
          
        for (int i = 0; i < n; i++) {  
            printRow(pascalTriangle.get(i));  
        }  
         
         
        for (int i = n - 2; i >= 0; i--) {  
            printRow(pascalTriangle.get(i));  
        }  
    }  
  
      
    public static void printRow(List<Integer> row) {  
        for (int num : row) {  
            System.out.print(num + " ");  
        }  
        System.out.println();  
    }  
}