WEEK 10

1)
```
import java.util.*
class vechicle{
    String regnumber;
    String ownerName;
    String vechicleTyne;
    Public Vechicle (String regnumber, String
    ownerName, String vechicle Tyne){
        this.regnumber.regnumber;
        this.ownername = ownerName;
        this.vechicleTyne = vechicleTyne;
}
```

2)
```
puplic class main {
    Pupsic static void main (String[] args){
        Scanner sc = new scanner(System.in);
        int n = sc.nextInt();
        sc.nextLine();
        TreeMap<character, integer> freq = new
        TreeMap<>();
        for(int i=0;i<n;i++){
            String line = sc.nextLine();
            for(int j=0; j< line.length();j++){
                char c = line.charAt(j);
                if(c==' ') continue;
                if ((c>='A' && A<='Z') ||
                (c>='a') && c<='z')){
                freq.put(c, freq.get ordefault
                (c,0)+1);
}
}
}
```

```
        else if (removeIndex < currentIndex) {
            currentIndex --;
        }
        else if (currentIndex >= playlist.size()) {
            currentIndex = 0;
        }
    }
} else if (comand.equals ("show")) {
    if (playlist.isEmpty ()) {
        System.out.print ("Empty");
    }
    else {
        for (String s : playlist) {
            System.out.print (s + "  ");
        }
    }
} else if (command.equals ("next")) {
    if (playlist.isEmpty ()) {
        System.out.print ("Empty");
    } else {
        currentIndex = (currentIndex + 1) %
                       playlist.size();
        System.out.print (playlist.get
                          (currentIndex) + "  ");
    }
}
}
```

```
Public class Main {
    Public static void main (String[] args) {
        Scanner sc = new Scanner (System.in);
        int n = sc.nextInt();
        LinkedList<String> playlist = new
        LinkedList<>();
        int currentIndex = 0;
    for (int i=0; i<n; i++) {
        String command = sc.next();
        if (command.equals("ADD")) {
            String song = sc.next();
            Playlist.add(song);
            if (Playlist.size()==1) {
                currentIndex = 0;
            }
        }
        else if (command.equals("Remove")) {
            String song = sc.next();
            int removeIndex = playlist.
                              indexOf(song);
            if (removeIndex != -1) {
                Playlist.remove(removeIndex);
                if (playlist.isEmpty()) {
                    currentIndex = 0;
                }
```

Week 9.

11)

```
public class Main {
    public static void main (String[] args){
        Scanner sc = new Scanner (System.in);
        int n = Integer.parseInt (sc.
                nextLine().trim());
        String [] input = sc.nextLine().
                trim().split(" ");
        ArrayList<Integer> result = new
                ArrayList<>();
        for (int i=0; i<n; i++){
            int num = Integer.parseInt
                (input[i]);
            if (result.isEmpty() ||
                num > result.get(result.size()-1))
                result.add(num);
        }
        System.out.println(result);
    }
}
```

```
Public class Main {
    Public static void validateFilename
    (String fileName) throws InvalidFileName
                                Exception {
        if (fileName.length()<3 ||
            ! fileName matches("[a-zA-Z0-9]){
            throw new InvalidFileNameException
            ("Error: Invalid File name.
              It must be alphanumeric and
              have a minimum length of 3
              characters.");
        }
    Public static void main (String[] args) {
    Scanner sc = new Scanner (System.in);
    String input = sc.nextLine();
    try {
        validateException (input);
        System.out.println("Valid
                            File name");
    }
    Catch (InvalidFileNameException e){
        System.out.println
        (e.getMessage());
    }
}
}
```

```
        int age = sc.nextInt();
        if (age < 18) {
            throw new InvalidAgeException("Age is
                                not valid to vote");
        }
        System.out.println("Eligible to vote");
    }
    catch (InvalidAgeException e) {
        System.out.println("Exception
            occured : InvalidAgeException: " +
            e.getMessage());
    }
    catch (InputMismatchException e) {
        System.out.println("An error occured: "+
        e.getClass().getName());
    }
    catch (Exception e) {
        System.out.println("An error occured: "+
                            e.getMessage());
    }
}
}
```

2)

```
class InvalidFileNameException extends Exception{
    public InvalidFileNameException(String
                                    message){
        super(message);
    }
}
```

```
}
if (username.length() < 6) {
    throw new InvalidUsernameException
("Invalid Username : Username must be
 at least 6 characters long");
}
}
Public static void main(String[] args){
    Scanner scanner = new Scanner(System.in);
    String username = scanner.nextLine();
    try {
        validUsername(username);
        System.out.println("username is valid: "+
                           username);
    }
    Catch (InvalidUsernameException e){
        System.out.println(e.getMessage());
    }
}
}
```

4)
```
class InvalidAgeException extends Exception {
    Public InvalidAgeException(String message){
        super(message);
    }
}
Public class Main {
    Public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
```

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int duration = scanner.nextInt();
    try {
        validateMeetingDuration(duration);
        System.out.println("Meeting
        scheduled successfully!");
    }
    catch (InvalidDurationException e) {
        System.out.println(e.getMessage());
    }
}
```

2)
```
class InvalidUsernameException extends Exception{
    public InvalidUsernameException(String
                                    message){
        super(message);
    }
}
public class Main{
    public static void validateUsername
    (String username) throws InvalidUsername
                             Exception{
    if (username.contains(" "))
        throw new InvalidUsernameException
    ("Invalid Username: Username cannot
     contain spaces");
```

```
Public double calculate
    double total = 0.0;
    for (int i=0; i< count; i++) {
        total += products[i].getValue();
    }
    return total;
}
}
```

5)
```
import java.util.Scanner;
interface AgeCalculator {
    int CalculateAge(int birthyear);
}
class HumanAgeCalculator implements
AgeCalculator {
    Public int CalculateAge(int birthyear){
        return 2024 - birthyear;
    }
}
```

2)

Week 8

1)
```
class DotException extends Exception {
    DotException(String msg){
        super(msg);
    }
}
```

```
        return price;
    }
}
class Produce extends Item {
    Public Produce(String name, double price)
        super(name, price);
    }
    @Override
    Public double calculateCost() {
        return price;
    }
}
```

Week 7

```
interface CostCalculator {
    void getEnergyDetails(Scanner scanner);
    void calculateAndDisplayCost();
}
class EnergyConsumptionTracker implements CostCalculator
    Private double rateProduct;
    Private int numDays;
    Private double[] dailyConsumption;
    EnergyConsumptionTracker(double rateperUnit,
                             int numDays) {
        this.rateperUnit = rateperUnit;
        this.numDays = numDays;
```

```
class cuboid {
    Protected double length, width, height;
    Public cuboid (double length, double width,
                   double height){
        this.length = length;
        this.width = width;
        this.height = height;
    }
    Public double calculateVolume(){
        return length * width * height;
    }
}
class cube extends cuboid {
    Public Cube (double side){
        super (side, side, side);
    }
    @ override
    Public double calculateVolume(){
        return length * length * length;
    }
}
class Item {
    Protected string name;
    Protected double price;
    Public Item (string name, double price){
        this.name = name;
        this.price = price;
    }
```

```
class Product {
    Public double price;
    Public Product (double price) {
        this.price = price;
    }
}
class DiscountedProduct extends Product {
    Private double discountRate;
    Public DiscountedProduct (double price,
    double discountRate) {
        super (price);
        this.discountRate = discountRate;
    }
    Public double calculateSellingPrice () {
        if (discountRate > 1) {
            return -1;
        }
        return price* (1- discountRate);
    }
}
class SalesTaxCalculator {
    Public static int calculateFinalPrice (int price,
                                           int taxRate)
        return price + (price * taxRate/100);
    }
    Public static double calculateFinalPrice (
                double price, double taxRate) {
        return price + ( price*taxRate/100);
    }
}
```

```
Public int getEnrollmentId()
    return enrollmentId;
Public String getStudentName()
    return studentName;
Public int getNumberOfSubjects(){
    return numberOfSubjects;
Public double calculateFee(){
    double registration = 1000;
    double subjectfee = numberOfSubjects * 800;
    double totalfee = registrationFee + subjectFee;
    if (numberofSubjects > 5)
        totalfee = totalfee - (totalfee * 0.20);
    return totalfee;
```

Week 6.

1) class PremiumSubscription {
    double baseMonthlyCost;
    double serviceTax;
    double extraFeatureCost;
    PremiumSubscription(double baseMonthlyCost,
    double serviceTax, double extraFeatureCost){
        this.baseMonthlyCost = baseMonthlyCost;
        this.serviceTax = serviceTax;
        this.extraFeatureCost = extraFeatureCost;
    }
    double calculateMonthlyCost(){
        return baseMonthlyCost + serviceTax +
        extraFeatureCost;
    }
}

```
Public int getBookingId() {
    return bookingId;
}
Public String getCustomerName() {
    return CustomerName;
}
Public double getDistanceTravelled() {
    return distanceTravelled;
}
Public double calculateFare() {
    double baseFare = 50;
    double distanceFare = distanceTravelled*
    double totalFare = baseFare + distanceFare;
    if (distanceTravelled > 20) {
        Totalfare = Totalfare - (totalfare * 0.10
    }
    return totalfare;
}

class Student {
    Private int enrollment id;
    Private String StudentName;
    Private int numberOfSubjects;
    Public Student (int enrollment, String
        StudentName, int numberOfSubjects) {
            this.enrollmentId = enrollmentId;
            this.StudentName = StudentName;
            this.numberOfSubjects = numberOfSubj
```

```
Public double getUnitsConsumed() {
    return unitsConsumed;
}
Public double calculateBill() {
    double bill = 0;
    double units = unitsConsumed;
    if (units <= 100)
        bill = units * 5;
    else if (units <= 200)
        bill = 100 * 5 + (units - 100)
    else
        bill = 100 * 5 + 100 * 7 + (units - 200) * 10;
    if (bill > 2000)
        bill = bill - (bill * 0.05);
    return bill;
}

class Booking {
    Private int bookingId;
    Private String customerName;
    Private double distanceTravelled;
    Public Booking (int bookingId, String customerName,
            double distanceTravelled) {
        this.bookingId = bookingId;
        this.customerName = customerName;
        this.distanceTravelled = distanceTravelled;
    }
```

```
Account acc = new Account (accountNumber,
                customerName, initialBalance);
acc.deposit(depositAmount);
acc.withdraw(withdrawalAmount);
customers[i] = acc;
}

for(Account acc : customers)
    System.out.println("Account Number" +
                acc.getAccountNumber());
    System.out.println(acc.getCustomerName());
    System.out.println(acc.getBalance());

Class Customer {
    Private int customerId;
    Private String customerName;
    Private double unitsConsumed;
    Public customer(int customerId, String
        customerName, double unitsConsumed) {
        this.customerId = customerId;
        this.customerName = customerName;
        this.unitsConsumed = unitsConsumed;
    }
    Public int getCustomerId() {
        return customerId;
    }
    Public String getCustomerName() {
        return customerName;
    }
```

```
else
    System.out.println(sb.toString());
```

5)
```
int T = Integer.parseInt(sc.nextLine());
for(int i=0; i<T; i++){
    String s = sc.nextLine();
    if(s.length()==4 && s.matches("[1-9]{3}")
        && !(s.charAt(0)==s.charAt(1) &&
    s.charAt(1)==s.charAt(2) &&
    s.charAt(2)==s.charAt(3)))
        System.out.println("yes");
    else
        System.out.println("no");
}
```

Week 5

1)
```
int n = Integer.parseInt(sc.nextLine());
Account[] customers = new Account[n];
for(int i=0; i<n; i++){
    int accountNumber = Integer.parseInt
                        (sc.nextLine());
    String customerName = sc.nextLine();
    double initialBalance = Double.parseDouble
                            (sc.nextLine());
    double depositAmount = Double.parseDouble
                           (sc.nextLine());
    double withdrawalAmount = Double.parseDouble
                              (sc.nextLine());
```

```
    }
  }
  if (firstRepeated != -1)
    break;
  }
}
if (firstRepeated != -1){
    System.out.println(firstRepeated);}
else {
  System.out.println("No repeated element
                      found in the array");
}
```

Week 4

1)
```
int n = Integer.parseInt(num.nextLine());
for(int i=1; i<=n; i++){
    String a = num.nextLine();
    int count=0, total=0, sum=0;
    for(char ch: a.toCharArray()){
      if (ch=='?')
        count++;
      else if (ch=='.')
        sum++;
      else if (ch=='!')
        total++;
    }
System.out.println(count+" "+sum+" "+total);
}
```

4)

```
int N = num.nextInt();
int M = num.nextInt();
int sum = 0;
int [][] arr1 = new int [N][M];
int [][] arr2 = new int [N][M];
for (int i=0 ; i<N ; i++) {
    for (int j=0; j<N; j++) {
        arr1[i][j] = num.nextInt();
    }
}
for (int i=0; i<N; i++) {
    for (int j=0; j<N; j++) {
        arr2[i][j] = num.nextInt();
    }
}
for (int i=0; i<N; i++) {
    for (int j=0 ; j<M ; j++) {
        sum = arr1[i][j] + arr2[i][j];
        System.out.print("It" +sum);
    }
}
System.out.println();
```

5)

```
int n = scanner.nextInt();
int[] arr = new int[n];
for (int i=0; i<n; i++)
    arr[i] = scanner.nextInt();
int firstrepeated = -1;
for (int i=0; i< arr.length ; i++) {
    for (int j=i+1; j< arr.length; j++) {
        if (arr[i] == arr[j]) {
```

1)
```
int n = num.nextInt();
int[] a = new int[n];
int m;
for (int i=0; i<n; i++)
        a[i] = num.nextInt();
Arrays.sort(a);
int sum = a[i] + a[n-3];
System.out.println(sum);
```

2)
```
int n = num.nextInt();
int sum1 = 0, sum2 = 0;
int[][]array = new int[n][n];
for (int i=0; i<n; i++){
      for (int j=0; j<n; j++){
            array[i][j] = num.nextInt();
      }
}
for (int i=0; i<n; i++)
        sum1 = sum1 + array[i][i];
for (int i=0; i<n; i++)
        sum2 = sum2 + array[i][n-1-i];
System.out.println(sum1);
System.out.println(sum2);
```

3)
```
int n = num.nextInt();
int[] arr = new int[n];
for (int i=0; i<n; i++)
      arr[i] = num.nextInt();
int sum = arr[0] + arr[n-1];
System.out.println(sum);
```

```
for (int i = rows-1; i >= 1; i--) {
    for (int j = 1; j <= i; j++) {
        System.out.print("*");
    }
    System.out.println();
}

for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n-i; j++) {
        System.out.print(" ");
    }
    for (int j = 1; j <= 2*i-1; j++) {
        System.out.print(j);
    }
    System.out.println();
}

int N = scanner.nextInt();
int count = 0;
for (int i = 1; i <= 9 && count < N; i++) {
    for (int j = 0; j <= 9 && count < N; j++) {
        for (int k = 0; k <= 9 && count < N; k++) {
            if (i != j && j != k && i != k) {
                int num = i*100 + j*10 + k;
                if (num%3 == 0)
                    System.out.println(num);
                count++;
            }
        }
    }
}
```

4)
```
int a = I.nextInt();
int b = I.nextInt();
double d = a;
for (int c = 0; c < b; c++)
    d -= d * 0.15;
if (d > 10000){
    System.out.println("high", d)
}
else if (d > 5000)
    System.out.println("medium", d)
else
    System.out.println("Low", d)
```

5)
```
int n = num.nextInt();
int m = n,
int count = 0, sum = 0;
do {
    int digits = m % 10;
    count++;
    sum = sum + digits;
    m = m/10;
} while (m > 0);
if (count == sum)
    System.out.println("matched");
else System.out.println("does not matched");
```

6)
```
for (int i = 1; i <= row; i++) {
    for (int j = 1; j <= i; j++) {
        System.out.print("* ");
    }
    System.out.println();
}
```

4)
```
int a = f.nextInt();
int b = f.nextInt();
double d = a;
for (int c=0; c<b; c++)
    d -= d * 0.15;
if (d > 10000){
    System.out.println("high", d)
}
else if (d>=5000)
    System.out.println("medium", d)
else
    System.out.println("Low", d)
```

2)
```
int n = num.nextInt();
int m = n;
int count=0, sum=0;
do {
    int digits = m % 10;
    count++;
    sum = sum + digits;
    m = m/10;
} while (m>0);
if (count == sum)
    System.out.println("matched");
else System.out.println("does not matched");
```

b)
```
for (int i=1; i<=rows; i++) {
    for (int j=1; j<=i; j++) {
        System.out.print("* ");
    }
    System.out.println();
}
```

Week 2.

1)
```
int a = num.nextInt(), b = num.nextInt();
int c = num.nextInt(), d = num.nextInt();
int e = num.nextInt();
int avg = (a+b+c+d+e)/5;
System.out.println(avg);
if (avg >= 50)
    System.out.println("The student has passed");
else
    System.out.println("The student has failed");
```

2)
```
int N = num.nextInt();
if (N%5 == 0)
    System.out.println(N + " is a multiple of 5");
else if (N%7 == 0)
    System.out.println(N + " is a multiple of 7");
else
    System.out.println(N + " is neither multiple
                        of 5 nor 7");
```

3)
```
double h = num.nextDouble();
double W = num.nextDouble();
double BMI = W/(h*h);
System.out.println(BMI);
if (BMI < 18.5)
    System.out.println("underweight");
else if (BMI > 18.6 && BMI < 24.9)
    System.out.println("Normal weight");
else if (BMI >= 25.0 && BMI < 29.9)
else
    System.out.println("
```

```
System.out.println(avg);
if (avg>a && b<avg)
  System.out.println()
else if (b<avg && c<avg)
  System.out.println()
else if (a<avg && c<avg)
  System.out.println()
else
  System.out.println()
```

9)
```
int a = num.nextInt();
int b = num.nextInt();
int c = num.nextInt();
int d1=b-a, d2=c-b;
if (d==d2)
  System.out.println("true");
else
  System.out.println("false")
```

10)
```
int x = sc.nextInt(), y = sc.nextInt(), z = sc.nextInt();
int d1 = sc.nextInt(), d2=sc.nextInt();
double rateA = 1.0/x, rateB = 1.0/y, rateC = 1.0/z;
double workD1 = d1 * (rateA + rateB + rateC);
double workD2 = d2 * (rateA + rateB);
System.out.println(workD1);
System.out.println(workD2);
double totalDone = workD1 + workD2;
double remaining = 1.0 - totalDone;
System.out.println(remaining);
```

Java Observation

Week 1

1)
```
int N = num.nextInt();
int M = num.nextInt();
int c = Math.abs(100-N), d = Math.abs(100-M)
if (c<d)
    System.out.println("The integer closer to 100 is" +
                        N + " with a difference " + c)
else
    System.out.println("The integer closer to 100 is"
                        + M + " with a difference" + d);
```

2)
```
int n = num.nextInt();
int m = num.nextInt();
boolean cond1 = (n>0 && m%3!=0);
boolean cond2 = (m>0 && n%3!=0);
if (cond1 || cond2)
    System.out.println("one of the integers is
    positive while the other is not divisible by 3");
else
    System.out.println("Neither of the integers
    meets the condition.");
```

3)
```
int n = num.nextInt();
double m = n
System.out.println(n);
System.out.println(m);
```

4)
```
int n = num.nextInt();
int m = num.nextInt();
int sum = n+m;
int product = n*m;
```

3)

```
Public class Main{
    Public static void main (String[] args){
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        TreeSet<Integer> seats = new TreeSet
                                    <>();

        for(int i=0; i<n; i++){
            seats.add (sc.nextInt());
        }
        int m = sc.nextInt();
        sc.close();
        if (seats.contains(m)){
            System.out.println
            (m+" is present!");
        }
        else{
            System.out.println(m);
        }
    }
}.
```