```
Message
"str.replaceAll(""[^a-zA-Z0-9]"", "" "")"
import java.util.Arrays;  int[] ch = new int[256]; Arrays.fill(ch, 0); //
Explicitly sets all elements to 0
// Clear the StringBuilder sb.setLength(0);
use  HashSet<Character> seen = new HashSet<>(); when dealing with
character rather than string builder
recusion  solved for array datasets
Use Indices or Slices:  Use indices or array slices to handle subarrays
in recursive calls. For example, you can pass the start and end indices
for the current subarray to the recursive function.
reverse of array can also be done by using concept of swapping where ,
start is swappend with end and both counter are adjusted to swap next
number
basic way to solve array for recusion problem  if(index==a.length) return
m; //change the below one as per problems m.put(a[index],
m.getOrDefault(a[index],0)+1); return sumarray(a,index+1,m);
the base case should be in top  if(index==a.length) { return l; }  or in
general too else if we have sme line defined before this line  we can get
indexout of bound problem
##for finding subset of sum of array  public static List<List<Integer>>
findSubsets(int[] a, int index, int sum, List<Integer> current) { //
Result list to store all valid subsets List<List<Integer>> result = new
ArrayList<>();  // Base condition: if we reach the end of the array if
(index == a.length) { if (sum == 0) { result.add(new
ArrayList<>(current));  // If sum is zero, add the current subset to
result } return result; }  // Include the current element in the subset
and reduce the sum if (a[index] <= sum) { current.add(a[index]);
result.addAll(findSubsets(a, index + 1, sum - a[index], current));
current.remove(current.size() - 1);  // Backtrack }  // Exclude the
current element and move to the next result.addAll(findSubsets(a, index +
1, sum, current));  return result; }
In Java, the addAll() method is used to add all elements from one
collection (such as a list, set, or another collection) into another
collection. It appends all the elements from the specified collection to
the end of the calling collection.
List<List<Integer>> result = new ArrayList<>();
List<List<Integer>> subsets = findSubsets(arr, 0, sum, new
ArrayList<>()); sending empty arraylist
adding list into another list  result.add(new ArrayList<>(current));
removing element from list  current.remove(current.size() - 1)
"list1.add(""Apple""); list1.add(""Banana"");  list2.add(""Orange"");
list2.add(""Grape""); list1.addAll(list2);"
Backtracking Step: After exploring all subsets including 2 and 3, the
algorithm removes the 3 and backtracks to explore other possibilities.
current = [2] Next Recursion: After removing 3, the algorithm explores
further possibilities (such as subsets that exclude 2), ensuring that it
looks at all possible subsets that sum to 7.
"map.remove(""2"");  // Removes the entry with key ""2"" (Banana)"
sb.deleteCharAt(5);  // Removes the character at index 5 (the comma)
Remember that in a 2D matrix matrix[i][j], i refers to the row index and
j refers to the column index.
p[j][i]=m[row-(i+1)][j]; --bring last row to zero  you are shifting
matrix by 90 clockwise  for anticlockwise ,p[i][j]=m[j][cols-(i+1)];
```

Clockwise 90° rotation: The element at position (i, j) in the original matrix moves to (j, n-i-1) in the rotated matrix. Counterclockwise 90° rotation: The element at position (i, j) moves to (n-j-1, i). rotation logic
for matrix , do in paper and see whats the logic for p[i][j]
for reversal across each row   p[j][i] = m[row - (i + 1)][j];
atleast try with 2 example for coming with logic, se do u have to traverse through rows or cols for cols ,
p[i][j] = m[row - (i + 1)][j];
write first before rigting code, imp logic and part of code so that in beginning u think clearly and no panic of any problem further in exam
right down imp , brief of class,interface question. use of which collection and basic logic like for loop , for recussion, for string ,matrix , queue , of exception handling - may be 10 min
n.concat(...) doesn't modify n. The concat method creates and returns a new string, but you aren't assigning it back to n fix is
n=n.concat(m.substring(pos,m.length()-1)).concat(m.substring(0,pos-1));
LinkedHashMap-Maintains insertion order. TreeMap-Keeps keys sorted (natural order or custom comparator).
similarily for set LinkedHashSet TreeSet
Set<Integer> linkedHashSet = new LinkedHashSet<>(Arrays.asList(1, 2, 3, 4, 5)); System.out.println(linkedHashSet); // Output: [1, 2, 3, 4, 5]
The line k = k % n; // Normalize rotation count ensures that the number of rotations k is adjusted to a valid range within the length of the string (n).
"for (int i = b - 1; i > a; i--) { System.out.print(i + "" ""); }
incremental other way round"
sometimes, when they ask for some output , like beep , with array of both , just print it . don't store in another array . its simple question
<Media omitted>
convert into integer a= (int) (remainder*Math.pow(10,count--));
Math.pow(10, 2) = 100.0
"System.out.print(""Enter a number: ""); int number = scanner.nextInt(); int reversed = 0; while (number != 0) { int digit = number % 10; // Extract the last digit reversed = reversed * 10 + digit; // Add it to the reversed number number /= 10; // Remove the last digit }"
String numStr = Integer.toString(number); convert number to strng to print digit
String s=Integer.toString(n); for(char c:s.toCharArray()) { System.out.println(c); }
always check for negation of what they ask if (digit % 2 != 0) { // Check if it is odd allEven = false; break; } boolean allEven = true; compare negation of condition
field[i] = random.nextInt(21) - 10; // Random numbers from -10 to 10
very imp-write logic and imp tricsk u are gonna use to solve each question of exam and then start coding because u could have time , exam phscological pressure in the last 20 min , may be brain wud not think in right direction ,so if u have written on paper ,just have to copy paste
after every question solution, check if result correct , something missing in 1 min
in last 10 min , check if all the submission correct
Difficulty Levels: Typically, exams start with easy or medium questions, and the difficulty increases. Prioritize solving easier questions first.

Here’s a comprehensive strategy to ace your Java coding exam, ensuring you stay calm, confident, and focused: 1. Understand the Exam Format Time Allocation: Divide the total exam duration by the number of questions. If it’s a 3-hour exam with 9 questions, you roughly get 20 minutes per question. Keep 15 minutes for reviewing. Difficulty Levels: Typically, exams start with easy or medium questions, and the difficulty increases. Prioritize solving easier questions first. 2. Prepare Strategically Before the Exam Master Core Topics: Strings: Manipulation, substrings, rotations, palindrome, and patterns. Arrays: Sorting, searching, prefix sums, sliding windows, subarrays, rotations. Recursion: Base cases, backtracking, subsets, permutations. Matrix: Traversals (row-wise, column-wise, diagonals), rotations, searching. Data Structures: ArrayLists, HashMaps, Stacks, Queues. Math: GCD, LCM, modular arithmetic, prime numbers. OOP Concepts: Encapsulation, inheritance, method overriding, and polymorphism. Practice Previous Papers: Identify patterns or recurring types of questions. Time yourself while solving these questions. Build Templates: Prepare reusable code templates for recursion, matrix traversal, sorting, searching, etc. Example: A skeleton for recursion or DFS in a 2D grid. Write Clean Code: Always include meaningful variable names and comments. Practice proper indentation for better readability. 3. During the Exam Phase 1: Analyze Questions (First 5–10 Minutes) Skim through all the questions. Classify: Divide questions into: Easy: Direct or familiar (e.g., array traversals, finding min/max). Medium: Needs some thought (e.g., recursion or combinatorics). Hard: Complex logic or time-consuming (e.g., advanced backtracking). Start with the easy ones to build confidence and momentum
####very imp Plan Before You Code: Quickly write down a pseudo-code or steps. Identify reusable parts (e.g., utility functions for matrix traversal).
###imp Stuck? Skip and Return Later: Don’t spend too much time on one question (e.g., more than 10 minutes). Move on and return later if time allows
Write comments explaining your logic if time runs out.
Infinite Loops in Recursion/Loops: Solution: Ensure base cases are correct and loops have termination conditions.
######## Final Exam Day Tips Relax and Rest: Get at least 7-8 hours of sleep before the exam. Stay Positive: Believe in your preparation. Avoid Overthinking: Focus on solving questions step-by-step. Carry Essentials: Ensure you have all necessary materials (e.g., pen, ID, etc.)
queue-eneque,deque,front,empty
empty queue has both rear and front as -1 , if both are equal that means first element when rear =size-1 then its full in enqeuing rear =rear+1
dequeue rear=rear-1
no wheil dequeing , front is front+1 till it equals -1 since dequeu is also done from head (front) ,first in first out
circula array
<Media omitted>
in link list , we only have one pointer called head, so to go to end of linked list we have to trsverse using temp or next pointer
<Media omitted>
Here, x does not store the actual data of these objects, but only references to them. In particular, this can be the null reference, for which one must watch out, for example
<Media omitted>

```
<Media omitted>
<Media omitted>
enqueue is inserting element at end of queue
<Media omitted>
<Media omitted>
<Media omitted>
<Media omitted>
public void enqueue(int data) { Node newNode = new Node(data); if (rear
== null) {  // If queue is empty front = rear = newNode; return; }
rear.next = newNode; rear = newNode; }
ArrayList<Integer> numbers = new ArrayList<>(); numbers.add(1);
numbers.add(2); numbers.add(3);  // Converting to an Integer array
Integer[] array = numbers.toArray(new Integer[0]);
q.contains(33) public boolean contains(Object obj)
public boolean contains(Object obj) { Elem position = start; while
(position != null) { if (position.getObject().equals(obj)) { return true;
} position = position.getNext(); } return false; }
position = position.getNext();
adding a new element or array or queue is using enqueue , traversing
through each element of queue
int[] arr = new int[size()];
converted into integer  arr[index++] = (Integer) position.getObject();
The loop stops when temp.getNext() is null, meaning the last element is
not processed in this loop. This works only if you don’t need to process
the last element.
a for loop ten times (a) first output the current value of x a means  run
a loop from 0 to 10 first print value of x and then update value of x
double x = 2.0;
46341
o find the smallest positive int number n that is so large that the value
n · n can no longer be represented as a int means to find this value , if
value goes beyond this it would be long
so this one (long) n * n > Integer.MAX_VALUE
<Media omitted>
in linked list   Node newNode = new Node(name, age); is based on
constructor of Node/elem
"Node current = front; System.out.println(""Name: "" + current.name + "",
Age: "" + current.age);"
"In case sequence contains numbers < 0 or >= word.length() return the
String ""Invalid Sequence""  if (index < 0 || index >= word.length()) {
return ""Invalid Sequence""; }"
matrix  int[][] m={{1,2,3},{2,3},{1,3,2}}
here we have 3 inner array which can be represented as
m[i] where if i=1 it gives first array of matrix , in this way cols of
matrix can be initlized dynamically like below  int[][] result = new
int[a.length][]; for (int i = 0; i < a.length; i++) { if (a[i] < 0) { //
Ignore negative sizes (could also throw an error) result[i] = new int[0];
} else { result[i] = new int[a[i]]; // Initialize inner arrays with a[i]
size } }
If a[i] < 0, it assigns an empty array (new int[0]) to the corresponding
row.
For a = {2, 3, -1, 4}:  result[0] → new int[2] result[1] → new int[3]
result[2] → new int[0] result[3] → new int[4]
```

"finding unique words using primitive logix // Split the string into words by spaces and convert to lowercase String[] words = s.toLowerCase().split(""\\s+""); // Variable to keep track of unique word count int uniqueCount = 0; // Array to mark words we've already seen boolean[] seen = new boolean[words.length]; // Loop through each word for (int i = 0; i < words.length; i++) { if (!seen[i]) { // Check if this word is already marked uniqueCount++; // Count it as unique // Mark all occurrences of this word for (int j = i + 1; j < words.length; j++) { if (words[i].equals(words[j])) { seen[j] = true; // Mark duplicates } } }"
In the class Evaluation, implement the method double categoryValue(String category) d returns the value of all products there array of products going to be iterated so in evaluation constructor need to initalize products
this one public class Evaluation { private List<Product> products; // Constructor to initialize the list of products public Evaluation(List<Product> products) { this.products = products; } }   is just like declaring 2 list and assigning list1 to list2 list<int>=new arral<> list2 list2=list1
This approach assumes that the caller (who creates the Evaluation object) will provide a valid List<Product> object when constructing an Evaluation instance.
"List<Product> productList = new ArrayList<>(); productList.add(new Product(""item1"", 10)); productList.add(new Product(""item2"", 20)); Evaluation evaluation = new Evaluation(productList);"
"evaluation.cateogryValue(""Electronics""); will call this method but product array is initlized by constructor"
"Only the string ""Electronics"" is explicitly passed to the method. The evaluation object itself is not passed as an argument but is implicitly available as this."
"one way evaluation evaluation = new evaluation(products); another way Restaurant r1 = new Restaurant(""The Gourmet Spot"", 4.5); Restaurant r2 = new Restaurant(""Tasty Treats"", 5.0); Restaurant r3 = new Restaurant(""Quick Bites"", 3.8); Restaurant[] restaurants = {r1, r2, r3};"
if (bestRestaurant != null) { , bestRestaurant is object here Restaurant bestRestaurant = finder.findBest(restaurants); array of object is passed , so even when u don't initilize object in constructot, it should be fine
Make sure that, as usual, the rating is in the interval between 0 and 5 inclusive. If a value is less than 0, set the rating to 0 and if the rating is greater than 5, set the rating to 5 it says handle this situation when initilizing the rating column , if you separately call it , then it won't handle all the time
in if condiiton, i have put this.rating which is wrong , i should check condition of rating from param and then assign value if(this.rating<0.0){ this.rating=0.0;} else if(this.rating>5.0){ this.rating=5.0;} else { this.rating=rating;} } ###very imp
instead of this String getname() { return this.name; } use String getname() { return name; } while setting , can use this but not while getting
Use this When There's Ambiguity You should use this if there is a local variable or parameter with the same name as the instance variable when same word is passed in param like in setting

```
way to handle null  if (restaurants == null || restaurants.length == 0) {
return null; // Return null if the array is empty or null }
both are similar  //Restaurant b=new
Restaurant(restaurant[0].name,restaurant[0].rating); Restaurant
b=restaurant[0];
way to compare variable of 2 object in class  Restaurant b=restaurant[0];
double rating=Double.MIN_VALUE; for(Restaurant r:restaurant) { //compare
if(r.getrating()>b.getrating()) }
Handling at Object Creation (Constructor) Why it's better: The
constructor ensures that as soon as the Restaurant object is created, the
rating is always valid.
Setter Version Purpose: Setters are used to update values after the
object is already created. so manually have to check and sset , which is
extra work
constructor initialize with null , give memory when nothing given
Use Case: The setter is useful when you need to modify the rating after
the object is created.
when generally they ask u to work with array of object of 1 class like
product , there would be another class which is of array type like we
have here product parser , then restaurantbig in live assign 5
"parts[0].matches(""\\d{5,}""))  // ID must have at least 5 digits"
"String[] parts = productString.split("";""); these condition if
(parts.length > 2 && !parts[2].isEmpty())"
###imp  DictionaryElement[] dictionary;  public ArrayDictionary(int size)
{ dictionary = new DictionaryElement[size]; } here ArrayDictionary is
class which contains array of DictionaryElement so each object of array
dict has DictionaryElement of some size each DictionaryElement is object
with key and value . so arraydic[i]=new DictionaryElement(,)
"### ArrayDictionary dictionary = new ArrayDictionary(3);  // Add some
elements dictionary.add(""apple"", 1);  boolean add(Object key, Object
value) { boolean flag = false; for (int i = 0; i < dictionary.length;
i++) { if (dictionary[i] == null || dictionary[i].getkey() == key) {
dictionary[i] = new DictionaryElement(key, value); flag = true; } }
return flag; } dicitionary is array, each index has dictinoary object"
when asked these variable is of object type so also declared as object
Object key; Object value;  public DictionaryElement(Object key, Object
value)
see what u are setting and returing in set and getter method, because of
copy paste can make mistake
boolean add(Object key, Object value): The methods searchs in the
dictionary for the next free position and then stores a DictionaryElement
there with the values key and value. If an entry with the key already
exists, overwrite the saved value vlaue with the new value.  for this
when not null then overrite rather than keepin gor condiiton and creating
new object
However, it may overwrite an existing value even if it was intended to
only update the value if the key already exists, depending on your use
case. This might not always be desirable if you wanted more granular
control.
for (int i = 0; i < dictionary.length; i++) { if (dictionary[i] == null)
{ dictionary[i] = new DictionaryElement(key, value); flag = true; } else
{ if(dictionary[i].getkey() == key) { dictionary[i].setvalue(value);
//here can directly return true; flag = true; } }
```

sometimes directly return true or false is better than using flag=true , which give smetimes diff result , so if get any issue directly return true or false
##imp create a PhoneBook class that contains only static class members this is not correct pb.add(name,number); when pb is of PhoneBookEntry pb.add(new PhoneBookEntry(name,number)); --correct
// Static list to store PhoneBookEntry objects private static List<PhoneBookEntry> phoneBook = new ArrayList<>();
You don't need to check for null in the list because an ArrayList (or most List implementations) shouldn't contain null unless explicitly added. ####important , in array declaration,may be we can find
"in this public static void main(String[] args) { // Test the PhoneBook functionality PhoneBook.change(""Alice"", ""123456""); } we directly call method using classname since all members of phonebook is static as mentioned in question Your taks is to create a PhoneBook class that contains only static class members. this means all are called using calssname"
this means we don't have constructor in phonebook
Yes, if all members of a class are static, there is no need for a constructor because constructors are used to initialize instance (non-static) members of a class. Since static members are tied to the class itself and not to any specific instance, you can work with them without creating objects of the class.
Static Variable: The static keyword indicates that the variable pb belongs to the class and not to any specific instance of the class. There is only one copy of the pb variable shared across all uses of the PhoneBook class, no matter how many times the class is referenced.
means same PhoneBookEntry to all
"try { n=Integer.valueOf(s); } catch(Exception e) { System.out.println(e.getMessage()); } gives For input string: ""rajan"""
"} catch (FileNotFoundException e) { // Handle the case where the file doesn't exist System.out.println(""Error: File not found.""); } catch (IOException e) { // Handle other IO-related exceptions System.out.println(""Error: An I/O error occurred.""); }"
Finally Block (finally): Always executes, regardless of whether an exception was thrown or not.
t the input is repeated so often (without producing further output), until for the first time while(true)--being used
###very imp to have break when asked for infinte loop Using while(true) in Java is appropriate when you need an infinite loop and don't know beforehand how many iterations are required. It allows you to repeatedly execute a block of code until a specific condition is met, at which point you can break out of the loop. However, it should always include a clear exit condition (e.g., using break or return), or it will lead to an infinite loop
"after continue in same lopp , nothing is reachable like here continue; //System.out.println(""i is ""+i);"
"in thi for(int i=0;i<10;i++) { if(i==4) { break; //System.out.println(""i is ""+i); } System.out.println(""i is ""+i); } if break print till 3 , if continue , skip printing when i =4 , so its like skipp that specific condiiton and then further continue executing loop"
when inheritng interface, use override and method be public when its public in interface

Integer integerObj = (Integer) obj;
Integer obj = 42; // Convert Integer object to int int value = obj.intValue();
instances of the interface - what does this mean In the given context, an instance of the interface refers to an object that implements the myDataInterface. However, it's important to note that interfaces in Java cannot be directly instantiated. Instead, a class that implements the interface is instantiated.
public myData() { this.data = new int[0]; // Initialize with an empty array } public void init(int[] array) { // Create a new array with the same length as the input array this.data = new int[array.length];
"@Override public int at(int n) { if (n >= 0 && n < data.length) { return data[n]; } throw new IndexOutOfBoundsException(""Index out of bounds""); }"
"In Java, once an array is created, its size cannot be changed. This means that you can't directly resize an existing array. However, you can create a new array with a different size and assign it to the variable (like this.data = new int[array.length];), effectively ""replacing"" the old array with a new one of the desired size. whichhas been done in above code"
Reassigning Arrays: You can change the reference of an array variable to point to a new array with a different size. This doesn't modify the existing array; instead, it creates a new array and assigns it to the variable, discarding the reference to the previous array.
In Java, the new keyword is used to create objects or allocate memory for variables,
"String str = ""Hello""; // This works because Java handles it internally. In this case, Java automatically handles memory for strings using a feature called string pooling, but it’s still creating the object in the background."
private ArrayList<Integer> data; // Constructor public MyData() { this.data = new ArrayList<>(); }
in copying copy one array data into instance variable of another class
to copy(!) the data of an int array to instances of the interface , it says just to a variable declared in myData class not of myData type variable
tips to solve matrix question do and traverse through matrix , seeing how to traverse thrugh row and oclumn using a[0][1], a[0][2]
when col changes, it goes through each row
if (n > 0 && n < 10)   0 ≤ n < 10
"System.out.printf(""%4d"", num)"
"for (int[] row : matrix) { for (int num : row) { System.out.printf(""%4d"", num); // Formatting for alignment } System.out.println(); }"
use the method Character.isDigit(char) to check if char is a digit.
"//this is problem   if(Character.isDigit(c)) { System.out.println(""digit is ""+c); sum+=c; System.out.println(""sum is in if ""+sum); } ccorrect to identify and sum digit is for(char c:str.toCharArray()) { if(Character.isDigit(c)) { //System.out.println(""digit is ""+c); sum+=Character.getNumericValue(c); //System.out.println(""sum is in if ""+sum); } }"
return str.contains(String.valueOf(sum));

```
String sumStr = String.valueOf(sum);  // Convert the sum to a string //
Check if the sum is contained in the input string return
str.contains(sumStr);
also explore function of character
"when its problem of string with muliple words  String[] chars =
str.split("""");  // Split the string into an array of characters  for
(String ch : chars) { if (Character.isDigit(ch.charAt(0))) { sum +=
Character.getNumericValue(ch.charAt(0)); } }"
char ch = '5'; int num = Integer.parseInt(String.valueOf(ch)); ###imp
sum+=Integer.valueOf(String.valueOf(c)); this is also fine
be careful of which datatype u are converting , if converting char , then
convert it into integer by using Character.
In this case, c is a char (character), which is a primitive type in Java.
The Integer.valueOf() method expects a String (or an int, but not a char
directly). Since c is a char and not a String, Java will not
automatically convert it to a String, and this results in a compile-time
error.
"In case no product with the name productName is found, return the null
reference means   searchs in the the given product array (Product[]) for
the first product means   Product search(String productName, Product[]
products)- in this products variable which is argument of method , is
here product array and each element of product is of product type which
was bit diff when we had diff class for array in which element beloned to
diff class . #####imp  Even though two products have the name
""Smartphone"", only the first occurrence (Samsung) is returned because
the method exits as soon as it finds a match"
return null;
either you have return or break but not both together  if
(p.name.equals(productName)) { return p; // break; }
if (product != null && product.getName().equals(productName)) { result =
product; break; // Exit the loop after finding the first match }
Elem first = x.getNext();         // Node 2 (original second node) Elem
second = first.getNext();    // Node 3 (original third node) Elem third =
second.getNext();    // Node 4 (original fourth node)  // Swapping
references to adjust the order x.setNext(second);    // x → Node 3
second.setNext(first); // Node 3 → Node 2 first.setNext(third);  ways to
swap location of linked list
These variables should only be accessed by the DictionaryElement class
itself means have getter and setter method in this class
public ArrayDictionary(int capacity) { dictionary = new
DictionaryElement[capacity]; size = 0; }  size is incremented after
inserting one value calling add method
private DictionaryElement[] dictionary;
live assign 51, q 3 is imp
in void method also , under some if condition , we can call return
keyword to end method .its better
imp string function -index of , substring , contains
startswith and endwith
"find no of cat in string  public static int countCat(String str) { //
Base case: If length of string is less than 3, ""cat"" can't exist if
(str.length() < 3) { return 0; }  // Check if the first three characters
are ""cat"" if (str.startsWith(""cat"")) { return 1 +
countCat(str.substring(1)); // Move one step ahead } else { return
countCat(str.substring(1)); // Move one step ahead } }  public static int
```

```
catfind(String s) {  int sum=0; if(!s.contains(""cat"")) { return 0; }
else {  sum++; int index=s.indexOf(""cat""); return sum +
catfind(s.substring(3+index));  }  }"
"if (str.substring(i, i + 3).equals(""cat""))"
int rollNumber = Integer.parseInt(details[2]);
define string array with months since for matrix asked which month has
max temp  so display month with using indexing(row of matrix is index of
month array)
"String[] months = {""January"", ""February"", ""March"", ""April"",
""May"", ""June"",  ""July"", ""August"", ""September"", ""October"",
""November"", ""December""};"
so in question lik this , add one more string array reprseting each row
of matrix
"System.out.printf(""%-10s: %.2f°C%n"", months[i], avgTemperatures[i]);"
gives this output January   : 32.20°C ######imp
%n:  This ensures a platform-independent newline (%n is preferred over \n
for cross-platform compatibility)  %-10s:  %s is a placeholder for a
string. -10 means the string will be left-aligned in a field of width 10
(if the string is shorter than 10 characters, spaces will be added to the
right).
"String formattedString = String.format(""%-10s: %.2f°C%n"", months[i],
avgTemperatures[i]); System.out.print(formattedString); ##########imp"
You deleted this message
<Media omitted>
What is the default value of a boolean variable in Java?  Answer: false
What is the difference between == and .equals() in Java?  Answer: ==
checks reference equality, while .equals() checks logical equality
ArrayIndexOutOfBoundsException
exceed the maximum value of an int (which is 2^31 - 1 or 2,147,483,647),
an overflow occurs
"##very imp The expression 5 + ""2"" will not result in an error, but
rather it will perform string concatenation. Here's why:  In Java, when
you use the + operator between a number (like 5) and a string (like
""2""), Java automatically converts the number to a string and
concatenates the two strings. So, 5 + ""2"" becomes ""5"" + ""2"", which
results in the string ""52""  better to run in intellij to get exact
result"
Summary of differences: a == b checks if both variables point to the same
memory location (reference comparison). a.equals(b) checks if the
contents of both strings are the same (content comparison).
"public class StringComparison { public static void main(String[] args) {
String a = new String(""rajan""); String b = new String(""rajan"");
System.out.println(a == b); // false (different memory locations)
System.out.println(a.equals(b)); // true (same content) } } ###imp If two
string objects are created dynamically (e.g., using new keyword), they
will not point to the same memory location"
"In Java, string literals are interned (they are stored in a common
pool). Since s1 and s2 refer to the same literal ""hello"", the ==
operator compares their references and returns true ##imp"
double a = 5 / 2; System.out.println(a);  2.0
"String a = ""rajan""; String b = new String(""rajan"");
System.out.println(a == b); // false (different memory locations)
System.out.println(a.equals(b));"
```

```
double a = 5 / 2.0; // One operand is a double System.out.println(a); //
Output will be 2.5
x++ is post-increment, so x++ returns 5 (the current value of x), and x
is incremented to 6. ++x is pre-increment, so it increments x to 7 and
then uses the value 7.
int i = 10; System.out.println(i++ + ++i); 22  i++ is post-increment, so
it first uses the value 10 and then increments i to 11. ++i is pre-
increment, so it increments i to 12 and then uses 12.
The array arr is initialized with 5 elements, all set to their default
value of 0 for int arrays
float f = 0.0f; double d = 0.0; System.out.println(f == d);  true
when u ovveride, make it public public String toString()
"@Override public String toString() { return ""Student{name='"" + name +
""', subjects="" + String.join("", "", subjects) + ""}""; }
Student{name='John Doe', subjects=Math, Science, History}"
"@Override public String toString() { return ""Student{name='"" + name +
""', subjects="" + String.join("", "", subjects) + ""}""; }
Student{name='John Doe', subjects=Math, Science, History}"
"when want to custom it then use stringbuilder public String toString() {
StringBuilder sb = new StringBuilder();
sb.append(""Student{name='"").append(name).append(""', subjects=["");
sb.append(""]}""); return sb.toString(); }"
constructor that initlializes list of movies means --and its code   map
or other collection question in class , map with different txpe of values
,practice  very imp  public Map<String, List<Movie>> getGenreMap() {
Map<String, List<Movie>> genreMap = new HashMap<>();  for (Movie movie :
m) { for (String genre : movie.getGenres()) { // If genre is not present,
add an empty list genreMap.putIfAbsent(genre, new ArrayList<>());  // Add
movie to the list for this genre genreMap.get(genre).add(movie); } }
return genreMap; }
imp to have this declaraton inside  Map<String, List<Movie>> genreMap =
new HashMap<>();#####
"for (Movie movie : entry.getValue()) { System.out.println(""   "" +
movie.getTitle()); }  keyset give keys, get value gives values and here
it iterate through each element of movie"
constuctor initlizes list means public StreamingService() { this.movies =
new ArrayList<>(); }
use  this.m=new ArrayList<>();
when using  genreMap.entrySet()
"for (Map.Entry<String, List<Movie>> entry : genreMap.entrySet()) {
System.out.println(""Genre: "" + entry.getKey()); for (Movie movie :
entry.getValue()) { System.out.println(""   "" + movie.getTitle()); }"
"Arrays.asList(""Thriller"", ""Crime"", ""Gangster"") --passing as list"
genreMap.putIfAbsent(genre, new ArrayList<>());  // Add movie to the list
for this genre genreMap.get(genre).add(movie);  adding list of movies for
genre
public static void static keyboard before return type
replace(seq,newseq) - this function is also imp where some char need to
be replaced
when getter and setter , keep variable as private
when says all variable to be externally accessible means variable need to
be declared public
in stirng question, think which function can easily solve this
exception handling custom one -- declaration imp
```

"when in question said , defined variable like left and right, should be able to store any data type available in java --very tricky question -- solution very imp public class Tuple<T, U> { private static int idCounter = 0; // Auto-incrementing ID private int id; private T left; private U right; // Constructor to initialize left and right values and assign a unique ID public Tuple(T left, U right) { this.id = idCounter++; this.left = left; this.right = right; } // Getters public int getId() { return id; } public T getLeft() { return left; } public U getRight() { return right; } // Setters public void setLeft(T left) { this.left = left; } public void setRight(U right) { this.right = right; } @Override public String toString() { return ""Tuple ID: "" + id + "" | Left: "" + left + "" | Right: "" + right; } }"
The id variable is static, but each object should have its own unique ID. You need an instance variable for id and increment the static counter each time an object is created. ########imp
Tuple<T, U> is a generic class declaration in Java. The T and U are type parameters that allow the class to work with multiple data types instead of being restricted to a specific one.
What is the Data Type of Object left;? In this declaration: java Copy Edit public class Tuple { Object left; Object right; } The variable left has the data type Object. In Java, Object is the superclass of all classes, meaning left can store any type of value (e.g., Integer, String, Double, List<Integer>, etc.). However, it lacks type safety, requiring explicit type casting when retrieving values. How is T left; Different from Object left;? When using generics like: java Copy Edit public class Tuple<T, U> { T left; U right; } T is a generic type parameter, which means the actual type is specified when creating an object. It preserves type safety by ensuring that only the correct type is stored and retrieved.
<Media omitted>
In Java, if you want to assign a unique ID to each object of a class, you can use a static counter that increments every time a new object is created. #######very imp
How ID is Assigned to Each Object? Declare a static variable (static int idCounter) to track the count of created objects. Each object gets a unique ID (id) which is assigned in the constructor. Increment the counter in the constructor to ensure the next object gets a new ID.
"Tuple<String, Integer> t1 = new Tuple<>(""A"", 10); because class is Tuple<T, U>"
"Without Generics (Using Object) java Copy Edit Tuple t2 = new Tuple(""A"", 10); // No <String, Integer> String s = (String) t2.getLeft(); // N"
"Using Generics (Type-Safe) Tuple<String, Integer> t1 = new Tuple<>(""A"", 10);"
"✅ Before Java 7 Tuple<String, Integer> t1 = new Tuple<String, Integer>(""Hello"", 100); Had to repeat <String, Integer> on both sides. ✅ After Java 7 (Using <>) Tuple<String, Integer> t1 = new Tuple<>(""Hello"", 100);"
"public Customer(String name, String street, String city) { this.name = name; this.street = street; this.city = city; } public CustomerParser(String line) { super("""", """", """"); // Call superclass constructor with empty values parse(line); // Parse the input line }"
"private void parse(String line) { String[] parts = line.split("",""); if (parts.length == 3) { // Ensure valid data this.name = parts[0].trim();

this.street = parts[1].trim(); this.city = parts[2].trim(); } else {
System.out.println(""Invalid input format!""); } all these variable of
super class are accessed by sub class"
"you can also call another method in constructor   public
CustomerParser(String line) { super("""", """", """"); // Call superclass
constructor with empty values parse(line); // Parse the input line } //
Method to parse a line in the format ""Name,Street,City"" private void
parse(String line) { String[] parts = line.split("",""); if (parts.length
== 3) { // Ensure valid data this.name = parts[0].trim(); this.street =
parts[1].trim(); this.city = parts[2].trim(); } else {
System.out.println(""Invalid input format!""); } } which initilize the
variable"
No, the subclass does NOT get a separate copy of the inherited variables
from the superclass. Instead, the subclass inherits the instance
variables from the superclass and shares the same copy
in case when each row has diff length of column avg[i] =
sum/temperatures[i].length; temp[i] is one row
"split(""\\s+"");"
when asked to be returing collection from checking something in already
another collection , initlialize collection and return it public
List<Post> collect(String keyword) { List<Post> collectedPosts = new
ArrayList<Post>(); for (int i = 0; i < posts.size(); i++) {   if
(posts.get(i).containsKeyword(keyword)) {
collectedPosts.add(posts.get(i)); }   } return collectedPosts; }
see collection question paper and also folder lookagain
Purpose: Adds all elements from another collection to the current
collection. Parameter: Takes a collection (e.g., List, Set, etc.) as an
argument, and it adds all the elements from that collection to the
current collection
"List<String> list1 = new ArrayList<>(); list1.add(""Hello"");
List<String> list2 = new ArrayList<>(); list2.add(""World"");
list1.addAll(list2);"
try to use hashset or treeset ,linked set when asking to find common
elements in 2 array
// Check which elements of the second array are in the HashSet for (int
num : arr2) { if (set.contains(num)) { common.add(num); set.remove(num);
// To avoid duplicates in the result } }
Set<Integer> set = new HashSet<>(); List<Integer> common = new
ArrayList<>();
"split(""\\s+"");"
imp :concat() doesn't modify the string: Strings in Java are immutable,
meaning concat() does not modify s. Instead, it returns a new string,
which you are not storing. Correct usage: s = s.concat(...)
"public static String encrypt(String msg, int shift){ char[] chword =
msg.toCharArray(); String res = "" ""; for(int i=0;i<
chword.length;i++){ res += (char)(chword[i]+shift); } return res; }
shfting by char"
when to check in string we have diff thing , use flag to convert once we
have digit or symbol boolean hasDigit = false; boolean hasAlpha = false;
boolean hasSpecial = false; if(pass.length<10){ return false; }
for(int i=0;i<pass.length;i++){
if(Character.isDigit(password.charAt(i))){ hasDigit = true; } else
if(Character.isAlphabetic(pass[i])){ hasAlpha = true; } else { hasSpecial
= true; } } return hasDigit && hasAlpha && hasSpecial;

"public class InvertNumberRec { public static int reverse(int num, int rev) { if (num == 0) return rev; return reverse(num / 10, rev * 10 + num % 10); } public static void main(String[] args) { int num = 5678; System.out.println(""Inverted Number: "" + reverse(num, 0)); // Output: 8765 } }"
public static int[][] createArray(int[] a){ int[][] res = new int [a.length][]; for(int i=0;i<a.length;i++){ res[i] = new int[a[i]]; } return res; }
if(tripsremaining>0){ tripsremaining--; }
"####new method   public Post01(String content, String hashtags) { this.content = content; String[] str = hashtags.split(""\\s+""); this.hashtags = new ArrayList<>(Arrays.asList(str)); }"
The constructor should store the hashtags as a list of individual hashtag strings
string to list
"for (Post01 post : posts) { for (String tag : post.getHashtags()) { hashTagCount.put(tag, hashTagCount.getOrDefault(tag, 0) + 1); } } hashTagCount.forEach((tag, count) -> System.out.println(tag + "": "" + count));"
sometimes ask you to run through 2 loops to work on map
if (countMap.get(num) >= 3) { return true; }
in startwith ,i can also put index startsWith(search, i) so can check some pattern in string
just using for(int i=0;i<a.length();i++) { int index=Integer.valueOf(String.valueOf(c.charAt(i))); System.out.println(index); for(int j=0;j<c.charAt(i);j++) charat(i) will give value of string , not its integer value , for that need to convert into integer for which character first need to converted into string and then to integer as done for index
int index2=c.charAt(i); gives This character is then implicitly converted to an int, which means it returns the ASCII
to display abc,123 to abbccc , using this for(int i=0;i<a.length();i++) { int index=Integer.valueOf(String.valueOf(c.charAt(i)));   for(int j=0;j<index;j++) { sb.append(a.charAt(i)); } } using content of one string as index of for loop
int index = Character.getNumericValue(c.charAt(i)); // Directly get numeric value
sb.append(String.valueOf(a.charAt(i)).repeat(index)); // Use repeat() for simplicity ####imp
String.valueOf(a.charAt(i)) converts the char to a String first, which is unnecessary when appending to StringBuilder, since StringBuilder.append(char) already handles char
replaceAll() java Copy Edit
Instead of (char) (i) + 'a', use (char) ('a' + i),
'a' + i → First, the character 'a' (ASCII 97) is incremented by i
public static boolean isPangram(String sentence) { sentence = sentence.toLowerCase(); // Convert to lowercase to handle case insensitivity HashSet<Character> letters = new HashSet<>(); for (char c : sentence.toCharArray()) { if (Character.isLetter(c)) { // Only consider letters letters.add(c); } } return letters.size() == 26; // A pangram must contain all 26 letters }
use (low.indexOf(letter) < 0) to check presence or absence of specific character

```
public Human(String name) { this.name = name; this.id = nextId++;
this.father = null; this.mother = null;  }  initlalize with null
private Man father; private Women mother; define in human which is
inherited by Man and woman subclass
public Human(String name, Man father, Women mother)
price = Double.parseDouble(priceAsString);
"if (parts.length != 3) { throw new IllegalArgumentException(""Invalid
input format. Expected format: 'ID;Name;Score'.""); }"
"(!score.matches(""\\d+(\\.\\d+)?""))"
"(isbn.isEmpty() || !isbn.matches(""\\d+""))"
isbn = Integer.parseInt(parts[0].trim());
"throwing new exception in try block  } else { throw new
RuntimeException(""Invalid operator. Expected '>' or '<'.""); } } catch
(NumberFormatException e){ throw new RuntimeException(""Invalid number
format.""); }"
"pictures++; System.out.println(""Taking picture with SLR Camera (Lens:
"" + lensName + "") - "" + this);  this and pictures which is static
variable is used in different class to where its declared"
when extedning abstract class  public Smartphone(String model) {
super(model, false); }
"SLRCamera slr = new SLRCamera(""Canon EOS"", ""50mm"");  public
SLRCamera(String model, String lensName) { super(model, true);  // Calls
the Camera constructor this.lensName = lensName;  // Sets the lensName
for the SLRCamera object }  public Camera(String model, boolean hasLens)
{ this.id = count++;  // Sets the id of the Camera object and increments
the count this.model = model;  // Sets the model field with the passed
model value this.hasLens = hasLens;  // Sets the hasLens field to true or
false based on the passed value }   When you call the constructor of
SLRCamera, it first invokes the constructor of the superclass Camera (via
super(model, true) When you create an SLRCamera object, the constructor
first calls super(model, true);. This calls the constructor of the
superclass Camera and assigns the model parameter to the model field of
the Camera class and sets the hasLens field to true.  The Camera
constructor is responsible for initializing the fields id, model, and
hasLens. The id is assigned an incrementing value using count++
(presumably for generating unique IDs for each Camera object),"
when using array to store object of diff class , count is counter which
is used to increase indexes of array and keep count of song in each of
songarray  like in this  public void addSong(Song song) { if (count <
songs.length) { songs[count++] = song; }  public SimplePlaylist(int
capacity) { songs = new Song[capacity]; count =0; }  public String
play(String title) { for (int i = 0; i < count; i++) { if
(songs[i].getTitle().equals(title)) { return songs[i].toString(); } }
this.learnedWords = new String[0];  // Empty array initially
"return String.format(""%02d:%02d:%02d"", hours, minutes, seconds);"
%02d means a two-digit integer (padded with zeroes if necessary)
s + sb.toString() + t
"// Method with 'throws Exception' but also handling an exception
internally public static void riskyMethod() throws Exception { try { int
result = 10 / 0; // This will cause an ArithmeticException
System.out.println(""Result: "" + result); } catch (ArithmeticException
e) { System.out.println(""Caught inside method: "" + e.getMessage()); }
// Throwing an exception explicitly throw new Exception(""Explicitly
thrown exception""); }"
```

```
we have both kind of throws in this method
"throw new Exception(""Explicitly thrown exception""); here Exception is
called rather than custom"
use of substring(0,mid) substring(mid) would be one way to find char in
strin g
when asked for pattern in string , may be try to use startwith function
of string . for string function, think of which of this string function
is more efficient in this quesiton , also think in way to solve the
recusion question
"in string eturn , return null or """" in base case and for int recturn
0"
"return ""a"" + g(a, a[i]);-- value of a[i] could be index in next
recusrion run"
in base case, also compare index value if
if <0  or > than str.length or remaining string lenght is less than
target string length so ,now no comarison is possible
jagged array has each  rows with diferent column lenght
int[][] jaggedArray = new int[rows][]; for each row jaggedArray[i] = new
int[columns];  #######imp
"public class JaggedArrayDemo { public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);  // Ask for the number of rows
System.out.println(""Enter the number of rows:""); int rows =
scanner.nextInt();  // Create a jagged array int[][] jaggedArray = new
int[rows][];  // Loop to create and fill each row for (int i = 0; i <
rows; i++) { System.out.println(""Enter the number of columns for row ""
+ (i + 1) + "":""); int columns = scanner.nextInt();  // Initialize the
sub-array jaggedArray[i] = new int[columns];  // Fill the sub-array
System.out.println(""Enter "" + columns + "" elements for row "" + (i +
1) + "":""); for (int j = 0; j < columns; j++) { jaggedArray[i][j] =
scanner.nextInt(); } }  jagged array"
break come out of loop and return come out of method
char shiftchr = (char) (ch[i] + shift );
"if(sequence[i]<0 || sequence[i]>=word.length()){ return ""invalid""; }
##when running recusrion or ehecking indees"
chword[sequence[i]]-- print char of string based on index from different
array
return new double[]{avg, maxtempmonth};- return array where each element
store different details
double maxtemp = a[0][0];
(int j = 0; j < temperatures[i].length; j++)  --when each row has
different column
new homw assign
"Map<Integer, Student> studentMap = new HashMap<>(); studentMap.put(101,
new Student(""Alice"", 101));"
"LinkedList<Student> studentList = new LinkedList<>();
studentList.add(new Student(""David"", 201));"
"for (Map.Entry<Integer, Person> entry : personMap.entrySet()) {
System.out.println(""ID: "" + entry.getKey() + "" -> "" +
entry.getValue()); }"
"Map<String, List<String>> studentSubjects = new HashMap<>();
studentSubjects.put(""Alice"", Arrays.asList(""Math"", ""Science"",
""History""));"
private List<List<String>> employeeGroups; // Nested list (list of lists)
public Department(String name) { this.name = name; this.employeeGroups =
new ArrayList<>(); }  employeeGroups.add(new ArrayList<>());
employeeGroups.get(groupIndex).add(employeeName)
for (List<String> group : employeeGroups) { if
(group.contains(employeeName)) { return true; }
// Method to check if an employee exists in any team public boolean
hasEmployee(String employeeName) { for (List<String> team : teamSet) { if
(team.contains(employeeName)) { return true; } } return false; }
"Map<String, List<String>> map = new HashMap<>();  // Add elements to the
map map.put(""fruits"", new ArrayList<>(Arrays.asList(""apple"",
""banana"", ""orange"")))"
"// Add another element to an existing list
map.get(""fruits"").add(""grapes"");--add to list"
"map.get(""fruits"")- give list so apply list function"
"Map<String, Set<String>> map = new HashMap<>();  // Add elements to the
map map.put(""fruits"", new HashSet<>(Arrays.asList(""apple"",
""banana"", ""orange"")));"
"for (Set<String> set : map.get(""fruits"")) { if
(set.contains(""banana"")) { found = true; break; } }"
"// Add a new item to a specific set inside the map
map.get(""fruits"").get(0).add(""pear"");"
"Map<String, Map<String, List<String>>> university = new HashMap<>();  //
Initialize departments university.put(""Computer Science"", new
HashMap<>()); university.put(""Mechanical Engineering"", new
HashMap<>());"
"university.get(""Computer Science"").put(""Data Structures"", new
ArrayList<>(Arrays.asList(""Alice"", ""Bob"", ""Charlie"")))"
"appleProducts.put(""Apple"", new HashMap<>());
appleProducts.get(""Apple"").put(""iPhone 15"", 999.99);"
"categoryMap.getOrDefault(""Fruits"", new List[3])[0].add(""Apple"");"
"Map<String, List<String>[]> categoryMap = new HashMap<>();  //
Initialize an empty array of lists for a category
categoryMap.put(""Fruits"", new List[3]); // 3 lists in an array  //
Ensure each index of the array is initialized as a List for (int i = 0; i
< categoryMap.get(""Fruits"").length; i++) {
categoryMap.get(""Fruits"")[i] = new ArrayList<>(); }"
this is custom exception example  class InvalidAmountException extends
Exception { public InvalidAmountException(String message) {
super(message); }
"public void validateUsername() throws InvalidUsernameException { for
(char ch = 0; ch < username.length(); ch++) { if
(Character.isDigit(username.charAt(ch))) { throw new
InvalidUsernameException(""Invalid username: Cannot contain numbers."");
} } System.out.println(""Username is valid: "" + username); }"
remove digit from character  for (char i = 0; i < input.length(); i++) {
if (!Character.isDigit(input.charAt(i))) { result += input.charAt(i); } }
for (char i = 0; i < input.length(); i++) { char ch = input.charAt(i); if
(ch >= 'a' && ch <= 'z') { result += (char) (ch - 32);  // Convert
lowercase to uppercase } else { result += ch; } }
"first repeated character  for(char c:s.toCharArray()) { if(ch[c]!=0) {
System.out.println(""first repeated char ""+c); break; } else { ch[c]++;
}"
List<Character> l=new ArrayList<>();
don't use set for anagram problem
```

```
List<Integer> list = numList.getNumbers();  int[]
arr=list.stream().mapToInt(i->i).toArray(); ##convertint list to array
cross check exception like , when choosing form recommended one ,
IllegalArgumentException ,match with what said in question
when they say id should start from 1000 , then  static int
idcounter=1000; int id;
int[] arr = list.stream().mapToInt(i -> i).toArray(); list.stream() →
Convert List<Integer> to a Stream .mapToInt(i -> i) → Convert
Stream<Integer> to IntStream .toArray() → Convert IntStream to int[]
You could achieve the same result manually using a loop:  java Copy Edit
int[] arr = new int[list.size()]; for (int i = 0; i < list.size(); i++) {
arr[i] = list.get(i); // Manually convert Integer to int }
String date , date is defined as string
setter and getter for variable which is dfined in super class ##imp
you have setter and getter for variables which is defined in class , not
some variable which is defined in super class
"@Override public void useTicket() { if (ridesLeft > 0) { ridesLeft--;
System.out.println(""Ticket used. Remaining rides: "" + ridesLeft); }
else { System.out.println(""No rides left! Ticket expired.""); } }  in
use method , decrement the variable value"
p.add(new Post(content, Arrays.asList(hashtag))); when one of the
variable of object is of list type
<Media omitted>
<Media omitted>
"//way to make in required form if asked  hashtag=""#""+hashtag+"" "";"
if you want unique keyword ,use set collection to store unique keywords
List<Poat> collect(String keyvord)
<Media omitted>
<Media omitted>
if asked to exactly match the keyword, split by # delieter and then store
into variale and iterate through each and check if exact keyword exist in
new list
<Media omitted>
may be sometimes run a small java code to see desired result like in
above check if by splitting by space give me string array with these
result
can also store in string array
"When you apply trim() directly after split("" ""), you likely encounter
an issue because split("" "") returns an array of strings"
when in exam , don't compile , write ur logic as comment , or if time
after all , write logic for difficult logic
"public void post2(String content, String hashTags) { // Convert hashtags
string into a list List<String> hashTagList = List.of(hashTags.split(""
"")); // Create Post object Post newPost = new Post(content,
hashTagList); // Add post to the list p.add(newPost); }  convert string
to list since one variable is of list type"
suppose list has these value {Goodluck,trier,university} if asked if list
contains good , it gives false but in question its said , even if it
matches partial like good , it should return it so for this case list
need to iterated for each element
"to store string value with ""good bad trier"" into list of content ,
first split into array string and then create list  String [] tags =
hashTags.split(""\\s+"");  List<String> tagList = new
ArrayList<>(Arrays.asList(tags));"
```

```
"List<String> tagList = List.of(hashTags.split(""\\s+""));   List.of"
when asked if given keyword in either of content of keyword , use
tostring method, which gives one string out of all variale, its
customized one, and then keyword can e searched in that string
String var=post123.toString().toLowerCase(); if
(var.contains(keyword.toLowerCase())) { matchingPosts.add(post123); }
"public Map<String, Integer> countKeywords() { Map<String, Integer>
wordCount = new HashMap<>();  for (Post post : posts) { // Split content
into words String[] words =
post.getContent().toLowerCase().split(""\\s+""); for (String word :
words) { word = word.replaceAll(""[^a-zA-Z0-9#]"", """"); // Remove
punctuation wordCount.put(word, wordCount.getOrDefault(word, 0) + 1); }
// Count occurrences in hashtags for (String tag : post.getHashTags()) {
tag = tag.toLowerCase(); wordCount.put(tag, wordCount.getOrDefault(tag,
0) + 1); } } return wordCount; }"
contain thf' J>a.SSed keyword keyword either in the post's ronrrnr
(co.ate.at) or in the list of ha.sh 11gs (hashTags). means check in
string return string variable of class
To do this. you mUst first createc a List<String> from string hashtag-
means convert string into list format ,likewise when asked to convert
into map or other collection or primitve datatype
"public void post2(String content, String hashTags) { // Convert hashtags
string into a map (count occurrences) Map<String, Integer> hashTagMap =
new HashMap<>(); String[] tags = hashTags.split(""\\s+""); // Split by
spaces  for (String tag : tags) { hashTagMap.put(tag,
hashTagMap.getOrDefault(tag, 0) + 1); }  // Create Post object Post
newPost = new Post(content, hashTagMap); }"
"Set<String> hashTagSet = new
HashSet<>(Arrays.asList(hashTags.split(""\\s+"")));"
. The previously implemented constructor should also be extended so that
a mother and a father can be passed and set accordingly  when asked to
extend public Human(String name) { this.id =
ID_GENERATOR.getAndIncrement(); this.name = name; this.mother = null;
this.father = null; }  // Constructor with name, mother, and father
public Human(String name, Woman mother, Man father) { this.id =
ID_GENERATOR.getAndIncrement(); this.name = name; this.mother = mother;
this.father = father; }
"this is imp ""    // Constructor with only name public Human(String
name) { this.id = ID_GENERATOR.getAndIncrement(); this.name = name;
this.mother = null; this.father = null; }"" where mother and father set
to null ,likewise can be done for other variable"
"Woman grandma = new Woman(""Anna"");"
"Woman mother = new Woman(""Emma"", grandma, grandpa);"
private List<Product> products = new ArrayList<>(); The list is
initialized directly at the time of declaration. This means that whenever
an object of Webshop is created, the list is already initialized, even
before the constructor runs. This approach ensures that products is never
null.
when initilized in constructor  The list is not initialized at
declaration, meaning it starts as null. It gets initialized inside the
constructor when a Webshop object is created.
"public String getDescription() { return ""Title: "" + title + "",
Author: "" + author + "", Year: "" + year; }  it can have multiple
variable being returned by method"
```

```
"@Override public String getDescription() { return super.getDescription()
+ "" containing "" + numberOfArticles + "" articles""; }  again calling
super inside overrding the same method #### important"
public void enqueue(String value) { if (head == null) { head = new
Elem(value); return; } Elem temp = head; while (temp.next != null) temp =
temp.next; temp.next = new Elem(value); }  when checking temp.next not
null stop at 2nd last node , then its iterated by 1 and then new node is
added
head = new Elem(value); assigned based on constructor
when something has to be done with value of node, traversing to be done
just check if node is null or not  while (temp != null) { if
(temp.value.equals(s)) return true; temp = temp.next; }
for queue , start from head with temp node
public void swap(String s1, String s2) { Elem temp = head, first = null,
second = null; while (temp != null) { if (temp.value.equals(s1)) first =
temp; if (temp.value.equals(s2)) second = temp; temp = temp.next; } if
(first != null && second != null) { String tempVal = first.value;
first.value = second.value; second.value = tempVal; } }
"public static Baby parseBaby(String input) { StringTokenizer token = new
StringTokenizer(input, ""{}:; ""); String name = """"; int weight = 0;
String birthdate = """";  while (token.hasMoreTokens()) { String key =
token.nextToken().trim(); if (key.equals(""name"")) { name =
token.nextToken().trim(); } else if (key.equals(""weight"")) { weight =
Integer.parseInt(token.nextToken().trim()); } else if
(key.equals(""birthdate"")) { birthdate = token.nextToken().trim(); } }
return new Baby(name, weight, birthdate); }"
public static String findT(String s) { int length = s.length(); if
(length % 2 == 0) { String t = s.substring(0, length / 2); if (s.equals(t
+ t)) { return t; } } return null; } string is concat of identical string
class Smartphone extends Camera { public Smartphone(String model, int
megapixels) { super(model, megapixels); }  camera is abstract class
class MedListElem { Medium medium; MedListElem prev; MedListElem next;
class has variable of its own type
public MedListElem(Medium medium) { this.medium = medium; this.prev =
null; this.next = null; }
class Mediathek { private MedListElem head; private MedListElem tail;
public Mediathek() { this.head = null; this.tail = null; }
"try { // Aufruf der exceptionTrigger-Methode exceptionTrigger(); } catch
(ArithmeticException e) { return 1; // Wenn ArithmeticException auftritt,
gebe 1 zurück }        } throw new NoProblemException(""No exception
occurred, something went wrong!"");"
switch (n) { case 1: throw new ArithmeticException();
"input = input.replaceAll(""[{}]"", """");"
"// Extraktion der Werte int id =
Integer.parseInt(attributes.get(""cnr"")); String name =
attributes.get(""name""); double value =
Double.parseDouble(attributes.get(""value""));  // Client-Objekt
erstellen und zurückgeben return new Client(id, name, value);"
// push-Methode (vorgegeben) public void push(int value) { top = new
Elem(value, top); }  // pop-Methode: Entfernt das oberste Element oder
gibt -123456 zurück public int pop() { if (top == null) { return -123456;
// Falls der Stack leer ist } int value = top.value; top = top.next; //
Entferne das oberste Element return value; }  // invert-Methode: Dreht
das Vorzeichen aller Elemente um public void invert() { Elem current =
top; while (current != null) { current.value = -current.value; //
Vorzeichen umdrehen current = current.next; } }
"public String toString() { StringBuilder sb = new StringBuilder(""[ "");
Elem current = top; while (current != null) {
sb.append(current.value).append("" ""); current = current.next; }
sb.append(""]""); return sb.toString(); }"
public static int countSubstring(String s, String t) { int count = 0; int
index = s.indexOf(t);  while (index != -1) { count++; index =
s.indexOf(t, index + 1); } return count; }
try (BufferedReader reader = new BufferedReader(new
FileReader(filename))) { String line; while ((line = reader.readLine())
!= null) {
public Student(String name, String fachbereich, Professor prof) {
this.name = name; this.fachbereich = fachbereich; this.prof = prof; if
(prof != null) { prof.addStudent(this); } }###########
"for (Student s : studenten) { if (zeit > 0) {
System.out.println(""Student "" + s.getName() + "" wird beraten."");
zeit--; } else { System.out.println(""Professor "" + name + "" hat keine
Beratungszeit mehr übrig.""); break; }"
poll() is used to retrieve and remove the first element from a Queue
Student s = list.poll();
if (arr[start] > 0 && arr[end] > 0) { start++; end--; }
public Product getMostExpensiveProduct(String s) { Product mostExpensive
= null; for (Product product : products) { if
(product.getName().equals(s)) { if (mostExpensive == null ||
product.getPrice() > mostExpensive.getPrice()) { mostExpensive = product;
} }
"Journal journal = new Journal(""Research Journal"", ""John Doe"", ""Some
Venue"", 2020, numberOfArticles); articles[0] = journal;"
this means q.sum(), q which is queue can access sum method . and q object
is queue on which sum peration will happen , so this object iwll have
required pointer and data like start,end and can access object , start
with Elem temp=start , and then iterate over queue , this queue has this
temp poiting to
public void append(int[] newData) { for (int num : newData) {
enQueue(num); } }
public static Queue concat(Queue q, Queue p) { Queue newQueue = new
Queue(); }an object is queue is created inside method which is in queue
class
array[(front + i) % capacity] == m
rear = (rear + 1) % capacity;
front = (front + 1) % capacity;
ques 2 of assign 10 , array implementation of queue
"double b = stack.pop(); double a = stack.pop(); double result;  switch
(token) { case ""+"": result = a + b; break; case ""-"": result = a - b;
break; case ""*"": result = a * b; break; case ""/"": if (b == 0) throw
new ArithmeticException(""Division by zero""); result = a / b; break;
default: throw new Exception(""Invalid operator""); }"
"String[] tokens = s.split(""\\s+"");  for (String token : tokens) {"
stack.push(result);
catch (AgeTooLowException | IllegalAddressException e) -- both exception
in one line
assign 12 , q3 imp point  when in question , asked extend constructor ,
sometimes just add functionality which asked in question like
```

, extend the constructor of the Person class so that an AgeTooLowException is thrown means handle this logic in constructor
"public IllegalAddressException(String message, String address) { super(message+"" ""+address); } sometimes when throwing custom exception , multiple variable can be asked to pass , likely handled in custom exception constructor with super(message,age) ##very imp extends by Exception extends Exception throw new AgeTooLowException(""Age must be greater or equal to zero:"",age); then we asked multiple exception to throw , Person(String name, int age, String address) throws NullPointerException,AgeTooLowException{}   for this return this.id+"" ""+this.name+"" ""+this.age+"" ""+this.address;, u can remove this part ."
NullPointerException should generally be handled differently because it's a more general error that often signals issues in the code logic rather than invalid input
Recommended Approach: Catch NullPointerException separately or don't catch it explicitly if you want it to propagate and handle it globally
The catch blocks must be ordered, with more specific exceptions before more general ones (though here AgeTooLowException and IllegalAddressException are custom exceptions and can go before the NullPointerException)
try { persons[i] = new Person(name, age, address); System.out.println(persons[i]); } catch (AgeTooLowException | IllegalAddressException e) { System.out.println(e.getMessage()); } catch (NullPointerException e) { System.out.println(e.getMessage()); } custom before nullpointer
You only have to test if the address string consists of two parts Streetname 12, 1234 City
"As message they pass the string ""Names are not allowed to be null"" to the exception constructor means throw new AgeTooLowException(""Age must be greater or equal to zero:"",age);"
handling exception in constructor while chekcing variable and if something wrong ,
Woman mother; Man father; public Human(String name) { this.name=name; this.id=idcounter++; this.father=null; this.mother=null; } public Human(String name,Man father,Woman mother) { this.name=name; this.id=idcounter++; this.father=father; this.mother=mother; } if variale is defined and not paased in contructor param , set it to nulls since it should be initialized if param not passed
Next, extend the class Human so that it stores the mother (Woman mother) and the father (Man father) of a human in addition to the name and an ID. The previously implemented constructor should also be extended so that a mother and a father can be passed and set accordingly. ###imp
For this part, it should be possible to create objects of the class Human without information about the parents. To do this, you must implement corresponding constructors ##without info of some specific variables, these variables are not passed in constructor familiarize yourself with nested objects
if 2 version of constructor are in superclass, it should also be handled in subclass
"String motherName = (mother != null) ? mother.getName() : ""Unknown""; String fatherName = (father != null) ? father.getName() : ""Unknown""; return ""Human{name='"" + name + ""', id="" + id + "", mother="" + motherName + "", father="" + fatherName + ""}""; }"

```
"private static final List<String> GENERAL_INGREDIENTS =
Arrays.asList(""Yeast Dough"", ""Tomatoes"", ""Mozzarella"",
""Oregano"");"
this.specialIngredients = new ArrayList<>(specialIngredients);
allIngredients.addAll(GENERAL_INGREDIENTS);
"public NotComparableException(Pet pet) { if (pet instanceof Dog) {
message = ""Exception: You can not compare cats and dogs""; } else {
message = ""Exception: No comparison possible""; }"
"public TooHeavyException(double weight) { super(""Exception: Dogs with
overweight don't go for walks""); this.overweight = weight - 15; }"
public int compareTo(Pet pet) throws NotComparableException { if
(this.getClass() != pet.getClass()) { throw new
NotComparableException(pet); } return Double.compare(this.weight,
pet.weight); }
List<> l= new ArrayList<>(specialIngredients); wya to by default assign
value to list
List<Integer> intList = new ArrayList<>(Arrays.asList(1, 2, 3, 4, 5))
charList.addAll(Arrays.asList('A', 'B', 'C'));
Set<Integer> intSet = new HashSet<>(Arrays.asList(10, 20, 30, 40, 50))
"Map<Integer, String> intStrMap = new HashMap<>() {{ put(1, ""One"");
put(2, ""Two""); put(3, ""Three""); }}"
You deleted this message
public void reverse() { Node prev = null, curr = head, next; while (curr
!= null) { next = curr.next; curr.next = prev; prev = curr; curr = next;
} head = prev; }
l1.next.next = new Node(5);
use of previous and current pointer to delete specific node from linked
list
super.balance
dynamic array  public UArray(int initCapacity) { internalArray = new
int[initCapacity]; size = 0; }  private void ensureCapacity() { if (size
>= internalArray.length) { int newCapacity = internalArray.length * 2;
int[] newArray = new int[newCapacity]; System.arraycopy(internalArray, 0,
newArray, 0, size); internalArray = newArray; }
double x = (int) 1.999  Result: 1.0
"while ((line = reader.readLine()) != null) { if (line.startsWith(""#""))
{ reader.close(); throw new Exception(""Invalid format: Comments found in
file.""); }"
<Media omitted>
inserting new node at specififc posiition  if (current != null) {
newNode.next = current.next; current.next = newNode; }
for (int i = 0; i < n - 1 && current != null; i++) { current =
current.next; } this is how u go to specififc position of linked list
using for loop and then perform operation
public void takePicture() { pictures++; }
public static int calculate(int n) { if (n == 0) { return 0; } else if (n
== 1) { return 1; } else { return calculate(n - 1) + calculate(n - 2); }
implementing without switch or for
if (s1.startsWith(s2) && s1.endsWith(s2)) { return s1.substring(len,
s1.length() - len); }
"try {   if (operator.equals("">"")) {   } else { throw new
RuntimeException(""Invalid operator.""); } } catch (NumberFormatException
e) { throw new RuntimeException(""Invalid number format.""); }  throwing
new exception inside try block and catching with in catch block"
```

swapping in java if (first != null && second != null) { String temp =
first.value; first.value = second.value; second.value = temp; }
super.withdraw(d);
"try { int isbn = Integer.parseInt(parts[0]); String title =
parts[1].trim(); String abstrct = parts[2].trim(); if (title.isEmpty()
|| abstrct.isEmpty()) { throw new IllegalArgumentException(""Titel oder
Abstract darf nicht leer sein.""); } System.out.println(""ISBN: "" +
isbn + "", Title: "" + title + "", Abstract: "" + abstrct); } catch
(NumberFormatException e) { throw new IllegalArgumentException(""Part1
muss eine gültige Zahl sein.""); }"
private Medium medium; private MedListElem next; private MedListElem
prev; public MedListElem(Medium medium) { this.medium = medium;
this.next = null; this.prev = null; }
Mediathek result = new Mediathek(this.name); MedListElem current = head;
To use the class MyClass from mypackage in another file: import
mypackage.MyClass;
"String text = ""hello123""; boolean isMatch =
text.matches(""\\w+\\d+"");"
"String[] parts = input.split("" "", 2); input.split("" "", 2) means: ""
"" → The delimiter is a single space. 2 → The limit parameter tells Java
to split at most once (creates at most two parts)"
return new String[]{matcher.group(1), matcher.group(2)}; --returing
string array
return (result != null) ? Arrays.asList(result) :
Collections.emptyList();   the way to return null list
Collections.emptyList(); ####imp
txt fle should be stored in root directory
<Media omitted>
import optimizers.address.AddressOptimizer; , importing class defined in
other package like optimizer
} catch (IOException e) { in file question
while ((line = reader.readLine()) != null) { AddressOptimizer optimizer =
new AddressOptimizer(line); List<String> result = optimizer.optimize(); }
"A valid postal code always has five digits (leading zeros are
permitted). A city name may consist of upper- and lower-case letters ('a'
to 'z' and 'A' to 'Z') or space characters ' ' or commas ',' or hyphens
'-' or dots '.' or round brackets '(' and ')' String regex =
""^(\\d{5})\\s+([a-zA-ZäöüÄÖÜß ,\\-.()]+)$""; Pattern pattern =
Pattern.compile(regex); Matcher matcher = pattern.matcher(input); if
(matcher.matches()) { return new String[]{matcher.group(1),
matcher.group(2)}; }"
variable must not be modifiable after its initial assignment means it
should be declared as final
For each address read, run the optimize() method (from AddressOptimizer).
If its result is empty, output "empty result". O means create object of
AddressOptimizer and call optimize AddressOptimizer optimizer = new
AddressOptimizer(line); List<String> result = optimizer.optimize();
\\s: Matches any whitespace character (spaces, tabs, etc.). \\d: Matches
any digit (0-9). matcher.group(1): If the match is successful, group(1)
returns the first captured group, which is the postal code (the 5
digits). The parentheses () create a capturing group, so this part of the
regex captures the postal code.
^: Anchors the match to the beginning of the string $: Anchors the match
to the end of the string

• AddressUtil must not be extensible or instantiable. It should be only available in its own package means u have to import this package to access method in this class

AddressUtil.splitAddress(this); ##passing same object --very imp

No Anchors (^ and $): Pattern can match anywhere in the string. The string no longer has to start or end with the postal code and city name in the correct format. The regex can now match a substring within a larger string, not necessarily the entire string

AddressUtil cannot be instantiated, meaning you cannot create an object of AddressUtil using the new keyword

Example: [aeiou] matches any vowel

[^] (Negated Character Class): Matches any character not in the set.

Example: [^0-9] matches any non-digit character.

"a? matches a or """""

a|b matches either a or b

<Media omitted>

<Media omitted>

see in question where they asked to define method and in which class , it would be easier

The provided class Evaluation contains the main method which first reads in the total number of persons (int) and then creates Person instances with the read in names and nationalities not separate class for main , also Person instances means object of person

List<Person> personList = new LinkedList<>();

map.size() will give no of distinct keys---##imp for question

return nationalityMap.getOrDefault(nationality,0); if no nationality then 0

m.put(p.nationality,m.getOrDefault(p.nationality,0)+1); pass same arg , i was making mistake by putting p in getordefault

byte and short converted into int before computation

<Media omitted>

<Media omitted>

<Media omitted>

"String[] parts = date.split(""\\."");"

If the delimiter is +, it must be escaped because + is a special regex character.

"String[] parts = input.split(""\\+"");"

"If using multiple delimiters, use regex with the OR operator (|): java Copy Edit input.split("",|;|\\+""); // Splits by comma, semicolon, or plus"

<Media omitted>

<Media omitted>

<Media omitted>

Why Use \\ Instead of \ in Java? If you want to use a backslash (\) inside a Java string, you must type it twice (\\). 🔹 Example: If you want to match a dot (.) literally in regex, you must write \\. in Java. Why? First \ → Escapes the second \ in Java (Java sees \\ as a single \). Second \ → Escapes the . in regex (so it is treated as a normal dot, not a special character)

"String[] parts = date.split("".""); 🚨 Problem: The . in regex means ""any character"", so this will split at every character, not just the dots."

char current = input.charAt(i); char next = input.charAt(i + 1); // If two consecutive characters are the same if (current == next) {

charCount.put(current, charCount.getOrDefault(current, 0) + 2);} when
comparing 2 consecutive char , also in map increment can be done with
value 2 for each occurence rather than just 1 , or any value n
You should also extend the output of the toString() method with the name
of the Space Station ,means override tostring() and call super.tostring()
alongwith this class content
The SpaceShip class should be extended with the same aspects as the
SpaceStation class, i.e. a name (String name)
class SpaceAsset { protected double[] coordinates; protected double
speed; // Constructor initializes coordinates with x, y, z public
SpaceAsset(double x, double y, double z) { this.coordinates = new
double[]{x, y, z}; this.speed = 0.0; // Default speed } imp how u
initilize int[] with given default variable ###very imp
rather than loopin, you can call   int[] d=new int[]{0,1,1};
System.out.println(Arrays.toString(d)); arrays.tostring()
which initializes the coordinate array with the given values
this.coordinates = new double[3]; this.coordinates[0] = x;
this.coordinates[1] = y; this.coordinates[2] = z;
there are question where array are declared with this.words = new
String[100]; // Fixed size array this.wordCount = 0; fixed size and then
in some method , element is added and its index is maintained like this ,
where the next element should be inserted to private int wordCount; if
(wordCount < words.length) { words[wordCount++] = newWord; }
// Find the second occurrence of s2 in s1 after the first occurrence int
secondOccurrence = s1.indexOf(s2, firstOccurrence + s2.length()); ####imp
// Find the second occurrence of s2 in s1 after the first occurrence int
secondOccurrence = s1.indexOf(s2, firstOccurrence + s2.length()); // If
the second occurrence doesn't exist, return null if (secondOccurrence ==
-1) { return null; } // Return the substring after the second occurrence
of s2 return s1.substring(secondOccurrence + s2.length());
int firstOccurrence = s1.indexOf(s2);
StringBuilder is designed for cases where strings need to be modified
frequently, as it avoids creating a new string object every time
public static int countSubstringOccurrences(String s, String t) { int
count = 0; int index = 0; while ((index = s.indexOf(t, index)) != -1) {
count++; index++; // Startet die Suche nach dem nächsten Vorkommen }
return count; }
"in a list l={""banana"",""apple"",""orange""] , when
l.indexOf(""apple""); , it gives 1"
"[apple, banana, cherry] int index = example.indexOf(""banana"");
example.add(index + 1, ""star""); gives [apple, banana, star, cherry]"
"example.add(""star""); , when simply add this , it add elem at end of
list [apple, banana, star, cherry, star]"
calculateSum(String filename) throws IOException
professor.addStudent(this);
"printint element of list MedListElem current = head; while (current !=
null) { sb.append(current.medium.toString()).append(""\n""); current =
current.next; } return sb.toString();"
/ (a) Method to get the last element that is strictly smaller than the
given value public IntElem getPrev(int value) { IntElem current = start;
IntElem previous = null; while (current != null && current.getValue() <
value) { previous = current; current = current.getNext(); } return
previous; }
IntElem current = start; IntElem previous = null;

"s.charAt(0) == 'b' && s.endsWith(""c"") chartat(0) or startsith and viceversa for endwith function"
try { isbn = Integer.parseInt(parts[0].trim()); } catch (NumberFormatException e) { isbn = -1; } in catch block we can put some logic too rather than just printing message
Use Double.parseDouble(s) when you only need a double primitive. Use Double.valueOf(s) when you need a Double object (e.g., storing in a List<Double>).
If a variable is defined in a superclass, you can create getter and setter methods in either the superclass or the subclass
"@Override public String getName() { return ""Dog's name is: "" + super.getName(); } @Override public void setName(String name) { System.out.println(""Updating dog's name...""); super.setName(name); } this is how u do"
arrays method Arrays.sort(array), Arrays.fill(array, value) - Fills an entire array with a specified value. Arrays.fill(array, startIndex, endIndex, value), Arrays.equals(arr1, arr2,Arrays.toString(array),Arrays.copyOf(array, newLength) - Creates a new array with the specified length. Arrays.copyOfRange(array, start, end), Arrays.stream(array),
replace(oldChar, newChar) - Replaces all occurrences of a character. replaceAll(oldString, newString) - Replaces all occurrences of a substring using regex.
"String.join(""-"", ""2024"", ""02"", ""06"");"
"String formatted = String.format(""Name: %s, Age: %d"", name, age); System.out.println(formatted); // Output: Name: Alice, Age: 25"
Arrays.copyOf(array, newLength
Arrays.asList(array)
import java.util.Arrays; int[] numbers = {1, 2, 3, 4, 5}; System.out.println(Arrays.toString(numbers)); // Output: [1, 2, 3, 4, 5]
"String[] names = new String[]{""Alice"", ""Bob"", ""Charlie""};"
int sum = Arrays.stream(numbers).sum();
"List<String> list = new ArrayList<>(Arrays.asList(""Apple"", ""Banana"", ""Cherry""));"
"list.add(""Date""); // Adds an element list.remove(""Banana""); // Removes an element list.set(1, ""Mango""); // Updates an element at index 1 String fruit = list.get(0); // Retrieves an element int size = list.size(); // Gets the number of elements boolean exists = list.contains(""Apple""); // Checks if an element exists Collections.sort(list); // Sorts the list Collections.reverse(list); // Reverses the list"
"boolean exists = map.containsKey(1); // Checks if a key exists boolean valueExists = map.containsValue(""Three"");"
"for (Map.Entry<Integer, String> entry : map.entrySet()) { System.out.println(entry.getKey() + "" -> "" + entry.getValue()); }"
int top = stack.pop(); // Removes and returns the top element int peeked = stack.peek(); // Returns the top element without removing it
"Word-to-Digit Conversion Step 1: Store a mapping of number words to digits (e.g., ""one"" → 1). Step 2: Split the input string into words. Step 3: Convert each word into its corresponding digit using the mapping. Step 4: Return the final number as a string. 2️⃣ Digit-to-Word Conversion Step 1: Store a mapping of digits to words (e.g., 1 → ""one""). Step 2: Convert each digit in the number to a word using the mapping. Step 3: Return the final word sequence. 🔹 Java Implementation java Copy Edit

```
import java.util.*;  public class NumberConverter {  // Maps for
conversion private static final Map<String, Integer> wordToDigit = new
HashMap<>(); private static final Map<Integer, String> digitToWord = new
HashMap<>();  // Static block to initialize maps static { String[] words
= {""zero"", ""one"", ""two"", ""three"", ""four"", ""five"", ""six"",
""seven"", ""eight"", ""nine""}; for (int i = 0; i <= 9; i++) {
wordToDigit.put(words[i], i); digitToWord.put(i, words[i]); } }  //
Method to convert words to digits public static String
wordsToDigits(String input) { String[] words =
input.toLowerCase().split(""\\s+""); // Split by space StringBuilder
result = new StringBuilder();  for (String word : words) { if
(wordToDigit.containsKey(word)) { result.append(wordToDigit.get(word));
// Convert word to digit } else { return ""Invalid input: "" + word; //
Handle invalid words } }  return result.toString(); }  // Method to
convert digits to words public static String digitsToWords(String input)
{ StringBuilder result = new StringBuilder();  for (char ch :
input.toCharArray()) { if (Character.isDigit(ch)) { int digit =
Character.getNumericValue(ch);
result.append(digitToWord.get(digit)).append("" ""); // Convert digit to
word } else { return ""Invalid input: "" + ch; // Handle non-digit
characters } }  return result.toString().trim(); // Trim extra space at
the end }  public static void main(String[] args) { // Example Usage
System.out.println(wordsToDigits(""one two three""));  // Output: 123
System.out.println(wordsToDigits(""five eight zero"")); // Output: 580
System.out.println(digitsToWords(""123""));            // Output: ""one
two three"" System.out.println(digitsToWords(""709""));            //
Output: ""seven zero nine"" } } 🛠 Explanation 1️⃣ wordsToDigits(String
input) Splits the string into words. Maps each word to its corresponding
digit. Handles invalid input (e.g., ""onetwo"" will return an error). 2️⃣
digitsToWords(String input) Extracts digits from the input. Converts each
digit into its corresponding word. Handles non-digit characters
gracefully. 🔹 Sample Input & Output Input (Words)  Output (Digits) ""one
two three"" 123 ""five eight zero""     580 ""seven zero nine""     709
Input (Digits)   Output (Words) ""123"" ""one two three"" ""709""
     ""seven zero nine"" ""580""  ""five eight zero"" 🚀 Additional
Enhancements Support for numbers beyond single digits (""twenty-one"" →
21). Handle invalid words like ""hello"". Add support for hundred,
thousand, million, etc. Would you like me to add support for larger
numbers like ""three hundred twenty-four"" → 324? 😊"
if condition for checking if map contains certain key or value  if
(wordToDigit.containsKey(word))###very imp
if (Character.isDigit(ch)) { int digit = Character.getNumericValue(ch);
when working with character and getting digit value from string
Character.isLetter(char ch) Purpose: Checks if the given character is a
letter (A-Z or a-z).
Character.isDigit(char ch)
Character.isWhitespace(char ch) Purpose: Checks if the given character is
a whitespace character (spaces, tabs, newlines, etc.).
Character.toUpperCase(char ch)
Character.isUpperCase(char ch),Character.isAlphabetic(char ch) Purpose:
Checks if the given character is alphabetic (i.e., a letter, A-Z or a-z)
```

Character.getNumericValue(char ch) Purpose: Returns the numeric value of the character (for digit characters, returns the digit value, otherwise returns -1
Character.toString(char c) Purpose: Converts the character into a Strin
equals(Object obj) - Compares this object with another, getClass() - Returns the Class object of the instance., getClass().getName() - Returns the fully qualified name of the class.
"Any characters that are not letters, digits, or symbols could be considered ""junk"""
You deleted this message
Pattern p = Pattern.compile(pattern); Matcher m = p.matcher(input); return m.matches(); -- return in boolean
if (m.find()) { return m.group(); // Return the matched string } --- return value
public static void main(String[] args) { --if not given , problem in code doesn't recognise keyword . so very imp
"\\w matches any ""word character"", which includes letters (both lowercase and uppercase), digits (0-9), and the underscore (_) \\s matches any whitespace character (spaces, tabs, line breaks)."
whitespace (\t)
^[^a-zA-Z0-9]$- says start and end with symbol
<Media omitted>
m.find(): This method attempts to find the next substring in the input string that matches the regular expression pattern. It returns a boolean value
In Java, m.group() is a method of the Matcher class (from java.util.regex). It returns the substring of the input that was matched by the regular expression
m.group() or m.group(0) → Returns the entire matched text
"String text = ""Date: 2025-02-06""; Pattern pattern = Pattern.compile(""(\\d{4})-(\\d{2})-(\\d{2})"");"
In Java, \ is an escape character inside strings. To represent a single backslash (\) in a string, we must write \\. Since \d is a regex pattern, in Java we need to write it as \\d in a string.
"System.out.println(input.matches("".*\\d.*"")); // Returns true (contains digits System.out.println(input.matches(""\\d+""));"
".*   Matches any sequence of characters after the digit."
"str.split(""[, .]+""); // Split by comma, space, or dot"
"str.replaceAll(""[^A-Za-z0-9]"", """")-remove special character"
(?=.*[A-Za-z]) -(?=...) is a positive lookahead, which means: It does not consume characters (it doesn't move the regex cursor forward). It only checks if a condition exists somewhere in the string. If the condition is not met, the regex fails (the match is rejected). + → Matches one or more occurrences of the preceding token. . → Matches any single character except a newline. .+ → Matches one or more of any character
(?=.*[A-Za-z]) and (?=.*\d) are lookaheads. Lookaheads do not consume characters; they just check if the condition is met.
Why Do We Need .+ in ^(?=.*[A-Za-z])(?=.*\d).+$? (?=.*[A-Za-z]) and (?=.*\d) are lookaheads. Lookaheads do not consume characters; they just check if the condition is met. This means that after the lookaheads, we still need a pattern that actually matches the string. .+ ensures there is at least one character after the lookaheads. If .+ were missing, the regex could match an empty string, which is not what we want.

A character class in regular expressions is a set of characters enclosed in square brackets []
In regular expressions, a period . is a special character that matches any single character, but when inside a character class, it loses its special meaning and matches just the literal period.
(...) (Parentheses - Grouping): The parentheses create a group. This allows us to combine multiple parts of the regular expression into a single unit and treat them as a whole. It also allows us to capture the matched part for later use
"[+\\-*/]: The | is the alternation operator, which means ""or"". This allows the regex to match either the first part (\\d+, one or more digits) or the second part ([+\\-*/], any of the arithmetic operators \\-: The - inside square brackets can be treated as a range indicator, which is why it must be escaped with a backslash (\\). In a regular expression, outside of square brackets, the + character has special meaning: it means ""one or more of the preceding element."" However, inside square brackets, + loses its special meaning and is treated as a literal plus sign"
we have \\- because else - can be used as range like a-z0-9 something like this
// Loop until no more occurrences of the substring are found while ((index = str.indexOf(sub, index)) != -1) { count++; index++; // Move the index forward by 1 for the next search }
to check substring - use indexof, (str.startsWith(sub, index)) , too
"// Convert List to a single String with space separator. stringList is list String result = String.join("" "", stringList);"
"// Find index of ""Banana"" int index = stringList.indexOf(""Banana"");"
"you can also throw exception in default or one of the case statment of switch default: throw new Exception(""Invalid operator"");"
"The method ""+-*/"".contains(token) checks if the string ""+-*/"" contains the token string"
from if else to case where else part goes to default section of case study
for (int i = 0; i < 5; i++) { switch (i) {}
The while loop is typically used when you don't know the number of iterations in advance but have a condition that must be satisfied. A for loop is used when the number of iterations is known, or can be determined Use a while loop when you have a condition that depends on something else that is not necessarily a counter, or if the loop should continue as long as some condition is true. ##imp
while loop to use -Indeterminate Iterations:,Waiting for a Condition to Change, You keep checking something repeatedly until it satisfies a certain condition,
"do { System.out.println(""Value of i: "" + i); i++; } while (i <= 5);"
"userInput.matches(""\\d+"")"
"String str = ""hello123""; // Regular expression to match alphanumeric string boolean isMatch = str.matches(""[a-zA-Z0-9]+""); // Output result if (isMatch) {"
x = 5*(count++) + count--; gives 7 , count++ incremental after 5*(count++) executed so does decrement when count-- is again added so , it will be like 5*(1)+2 since 1 is incremented to 2 after (5*1)
"Yes, can use find() for substring matching- for matcher class but not avaialble in str.matche("""")"
b = a << 2; Shifts all bits in a to the left by 2 positions

```
int a = -8; // Binary: 1111 1000 (32-bit representation) int d = a >>> 2;
finding no of substring in string  while ((index = s.indexOf(t, index))
!= -1) { count++; // Increment count for every occurrence index +=
t.length(); // Move the index to the end of the current match }
public Queue(int capacity) { this.capacity = capacity; array = new
int[capacity]; front = size = 0; rear = capacity - 1; }  ###for queue
rear = (rear + 1) % capacity; array[rear] = item; size++; ##adding item
at rear end
in dequeue  int item = array[front]; front = (front + 1) % capacity;
size--;
You deleted this message
"public void push(int item) { if (top == capacity - 1) {
System.out.println(""Stack is full""); return; } array[++top] = item; }
public int pop() { if (top == -1) { System.out.println(""Stack is
empty""); return -1; } return array[top--]; }"
public static void generateRandomIntegers(String fileName, int n) throws
Exception { try (BufferedWriter writer = new BufferedWriter(new
FileWriter(fileName))) {  no catch block here , try in () but
initilizination where declaring method
writer.newLine();
try (BufferedReader reader = new BufferedReader(new
FileReader(fileName))) { String line; while ((line = reader.readLine())
!= null) {}
// Generic method that can accept any type of array public static <T>
void printArray(T[] array) { for (T element : array) {
System.out.println(element); }
index = Arrays.binarySearch ( anArray, 4711); yields index of the element
in anArray which contains the value value
IOException in file operation
while ((line = reader.readLine()) != null)  , whole (line=...) is in
close bracket and compared with null
Convenient Methods: BufferedWriter.newLine() helps in writing lines
properly, and BufferedReader.readLine() makes reading more efficient.
"capacity  The maximum number of elements the queue can hold (fixed size
of the array). size   The current number of elements in the queue (how
many elements are present)"
queue=new int[capacity]; --initilize in constructor
"throw new RuntimeException(""Queue is empty""); common eception"
int midIndex = (front + (size / 2)) % capacity;
queue.reverse2(queue2) passing original queue object which call method
if(queue[(front+i)%capacity]!=queue[(rear-i)%capacity]) -comparing front
to rear point
queue2.enqueue(queue[(rear - i) % capacity]);- iterating through element
from rear to fron t
while (size > 0) { int item = dequeue(); reversed[index++] = item; } ---
to reverse queue element
int poppedValue = stack[top--]; stack[++top] = value;
int [] [] b = new int [5][];
for (int i = 0; i < c.length; i++) 12 d [i] = (int[]) c[i].clone();
"String st = ""This is a String""; 4 StringTokenizer tk1 = new
StringTokenizer (st); 5 / / breaks String down into tokens ""This"",
""is"", ""a"", and ""String"""
```

"String st2 = ""17, 25, 77; 34 19""; StringTokenizer tk2 = new
StringTokenizer(st2, "" ,;""); while (tk2.hasMoreTokens()) {
System.out.println(tk2.nextToken()); }"
##im when filling another list from a list , former list need to emptied
after each entry to other list The easiest way to remove all elements
from a list is by using the clear() method
list.removeAll(new ArrayList<>(list));
for somehow l.clear didn't wrk , so to way is use removeall or initlize
list again
"finding of sum pairs for(int i=0;i<a.length;i++) { rem=target-a[i];
System.out.println(""for i ""+i+ "" rem no to find is ""+rem); for(int
j=i+1;j<a.length;j++) { if(a[j]==rem) { System.out.println(""rem value
matching at j ""+j +"" "" + rem); l.add(a[i]); l.add(a[j]); l2.add(l);
//l.removeAll(new ArrayList<>(l)); l=new ArrayList<>(); } }"
why clear doesn't work The issue in your code is that l is a reference
to the same list object throughout the loop. When you clear l or remove
its elements, you are modifying the same list object inside l2, since l2
holds references to l and not a new list
Without creating a new list, all references in l2 would point to the same
modified list, leading to empty lists in the final output.
using set to solve pairs , it insert first unseen value and then in end
when it find complement of number ,add to pairs
"Set<Integer> seenNumbers = new HashSet<>(); System.out.println(""Pairs
with sum "" + target + "":""); for (int num : arr) { int complement =
target - num; if (seenNumbers.contains(complement)) {
System.out.println(""("" + complement + "", "" + num + "")""); }
seenNumbers.add(num); }"
to find common character between 2 string , insert one string into set1
and then iterate through next string , and check if set1 contains
character and add into another list
we can also have List<StringBuffer> l3 = new ArrayList<>(); which can be
used to store character into stringbuffer and store in list
"for (int i = 0; i < s1.length(); i++) { for (j = 0; j < s2.length();
j++) { while (i<s1.length()&&j<s2.length()&&(s1.charAt(i) ==
s2.charAt(j))) { System.out.println(""value of i is ""+i+"" has
character ""+s1.charAt(i)); System.out.println(""value of j is ""+j+""
has character ""+s2.charAt(j)); l.add(s1.charAt(i));
sb.append(s1.charAt(i)); i++; j++; System.out.println(""here""); }
System.out.println(""value of i ""+i); // j=0;   } if (!l.isEmpty()) {
l2.add(l); l3.add(sb); l = new ArrayList<>(); i--; sb=new StringBuffer();
//j=0; } }###imp logic"
"sometimes comparing one string to concat of other 2 can solve result
like rajansah equals ""Rajan""+""sah"" or sah+rajan"
boolean result = str1.startsWith(str3) && str1.endsWith(str2); --use this
function too
one finger tips , have all string function and arrays function
To find the point of rotation, we need to check where s2 is a rotated
version of s1., s1=rajansah s2=sahrajan s3=s1+s1, find if s2 is in s3
String doubledS1 = s1 + s1; // Find the index of s2 in the concatenated
string int index = doubledS1.indexOf(s2); return index;
Non-static variables are stored in the memory area of an object
A ’static’ variable is stored in the memory area of the class
"result = numerator + ""/"" + denominator;"

"In Java, a wrapper class is a class that ""wraps"" or encapsulates a primitive data type into an object"
final, it means that the class cannot be extended or subclassed. In other words, no other class can inherit from a final class.
The type of Object here is Object, which is the root class of the Java class hierarchy. In Java, every class (except null) inherits from Object
The method setObject(Object newObj) accepts any type of object (as long as it's not null), and assigns it to the instance variable obj
String s = (String) x.getObject(),
", x.setObject(""text"")"
private Elem next; is defined inside Elem class
"When you define private Elem next; inside the Elem class, it means that each instance of the Elem class has a reference to another Elem object. This is typically used in data structures like linked lists, where each element (or ""node"") contains a reference to the next element in the list"
This is the name of the class, and it indicates that next is going to reference another object of the Elem type (which means each element of the list can point to the next element)
private int value; // value of the current node private Elem next; // reference to the next node in the list
public Elem(int value) { this.value = value; this.next = null; // initially, no next element }
@Override public String toString () { return obj.toString (); }
"while (position != null) { 21 str += position.getObject().toString() + "" ""; 22 position = position.getNext(); 23 }"
Elem newElem =new Elem(newObj),   public void push 12 (Object newObj) { 13 Elem newElem = 14 new Elem(newObj); 15 newElem.setNext(top); 16 top = newElem; 17 }
public Object pop () { 4 if (top != null) { 5 Elem temp = top; 6 top = top.getNext(); 7 return temp.getObject(); 8 } 9 e
If the parent class has a no-argument constructor, Java automatically inserts super(); at the beginning of the child class constructor.
If the parent class does not have a no-argument constructor, and you do not explicitly call super(args), a compilation error will occur.
Abstract classes cannot be instantiated, they only serve to construct subclasses
return this.getClass().toString(); 15 / / name of the class of an object, as String
Quadrilateral ql = new Quadrilateral(new Pt(0,0),new Pt(1,0), 5 new Pt(1,1), new Pt(0,1)); where each element of constructor is of another class
you can have constructor in abstarct class but it can not be initialted, only subclass can be initialted
"Path path = FileSystems.getDefault().getPath("".""", name); List<String> lines = Files.readAllLines(path, Charset.defaultCharset());"
BufferedReader nbr = Files.newBufferedReader(path, Charset.defaultCharset());
BufferedWriter nbw = Files.newBufferedWriter(path)
final method won't be extended or inherited
"logic of this ""all other samples will also be contaminated. In this case the method test() must change the status boolean tainted of each sample in Sample[] samples to tainted = true."" use of break in this"

```
"""  boolean test() { boolean flag=true; for (int i = 0; i <
samples.length; i++) { if (samples[i].tainted == false) { flag= false;
//break;  } samples[i].tainted=true; } return flag;  }"""
@Override public boolean test() { boolean allClean = true;  // Check for
contamination in each sample for (int i = 0; i < samples.length; i++) {
if (samples[i] != null && samples[i].tainted) { // If the sample is
contaminated, mark all samples as tainted for (int j = 0; j <
samples.length; j++) { if (samples[j] != null) { samples[j].tainted =
true; } } allClean = false;  // The experiment is not successful break; }
}  // If all samples are clean, the test was successful return allClean;
}
@Override void learn(String newWord) { for(int i=0;i<100;i++) {
if(s[i]==null) { s[i]=newWord; count++; break; } } } ,insert new word and
break
"Why Can't Static Methods Be Overridden? Static methods are bound to the
class and not to any specific instance. What Happens When You
""Override"" a Static Method? When a subclass defines a static method
with the same name and parameters as a static method in the parent class,
it hides the parent class's static method. You can declare a variable of
the parent class type and assign it to an object of a subclass. This
allows you to call methods defined in the parent class, but if the method
is overridden in the subclass, the subclass's implementation will be
executed. you cannot directly access members of the subclass that are not
part of the parent class. To access subclass-specific methods or fields,
you need to cast the reference to the subclass type.  The line Child
child = (Child) childAsParent; involves type casting in Java,
specifically downcasting from a parent class reference to a child class
reference.  Person[] people = new Person[2];  // Initializing the array
people[0] = new Person(""John"", 25); people[1] = new
Person(""Alice"", 30)  StringBuffer insert(): Inserts the given text at a
specified position in the current StringBuffer object. capacity():
Returns the current capacity of the StringBuffer object. // Insert text
at a specific position,  sb.insert(6, ""Beautiful ""); sb.delete(6, 17);
// Removes ""Beautiful ""  Validation: You can add logic inside the
setter method to ensure the validity of the values being assigned.  //
Constructor with two parameters public Car(String model, int year) {
this.model = model; this.year = year; }  // Constructor with only one
parameter, calling the other constructor public Car(String model) {
this(model, 2020); // Calling the constructor with two parameters }
content.append(line).append("" "");  nextLine(): Reads the entire line
(useful for sentences or strings with spaces). nextInt(): Reads an
integer. nextDouble(): Reads a double (floating-point number). next():
Reads a single word (stops at space).  int intValue =
Integer.parseInt(number); // Primitive,   Integer intObject =
Integer.valueOf(number)  String result = String.join("", "", ""apple"",
""banana"", ""cherry"");  if (obj instanceof Integer) {
System.out.println(""The variable is of type Integer."");
sb.getClass().getName(),        String s=sc.next(); sb.append(s+""\n"") -
-for leaving one line System.out.format(""%.2f"",pi) ;
System.out.format(""The number is: %5d\n"", number);  // Padded with
spaces to a width of 5  while (true) { System.out.print(""Enter student's
age (must be between 18 and 100): ""); this.age = sc.nextInt(); }  for
(int i = 0; i < matrix.length; i++) { // matrix.length gives the number
of rows for (int j = 0; j < matrix[i].length; j++)
```

```
System.out.print(matrix[i][j] + ""\t"");--for leaving tab space
if(ma[i][j]>max) { secmax=max; max=ma[i][j];  } else if(ma[i][j]>secmax)
{ secmax=ma[i][j]; }"
"list.containsAll(Arrays.asList(""apple"", ""banana"") Checks if the
collection contains all elements of another collection Checks if the
collection contains a specific element. list.contains(""apple"") →
Returns true list1.equals(list2), List<String> difference = new
ArrayList<>(list1);  difference.removeAll(list2); // Elements in list1
but not in list2 List<String> common = new ArrayList<>(list1);
common.retainAll(list2); // Common elements secondary diagonal (j == n -
i - 1) if(a>b) { if(a>c) { max=a;  }  The charCount array is of size 256
(new int[256]), so valid indices should be in the range from 0 to 255.
If ch is a primitive char, the toString() method might not work as
expected because char is a primitive type and doesn't have its own
toString() method. Calling toString() on a primitive type doesn't give
you the character's value in string form.  If ch is a Character object,
the toString() method works as expected and returns the
character as a string.  // Concatenate the characters to form the string
String result = """" + c1 + c2 + c3;  char[] chars = {'r', 'a', 'j'};  //
Use String.valueOf() to convert char array to String String result =
String.valueOf(chars);  // Loop through each character and convert to
String for (char c : chars) { result += Character.toString(c); }   for
(int i = 0; i < game.length; i++) { System.out.print("" | ""); for (int j
= 0; j < game[i].length; j++) {  sometimes , the array element become
limit of inner loop  There is an administrator password that applies to
all objects of this class. means there is variable which
store admin password  if some variable across all object , should be
static  for (int i = 0; i < entries.length; i++) { if (entries[i] ==
null) { entries[i] = new Entry(technicalTerm, explanation); break; }
break is imp since in one call only one object filled  public String
toString() { StringBuilder sb = new StringBuilder(); for (Entry entry :
entries) { if (entry != null) {
sb.append(entry.toString()).append(""\n""); --adding into one string ,
also calling tostring() //of other class } } return sb.toString(); }
This array is to be created in the constructor (containing 1000 null
references)  public TechnicalDictionary(Scanner scanner) { this.scanner =
scanner; this.entries = new Entry[1000]; // Initialize array with null
references }  public void addAssistant(Assistant assistant) { if
(assistants.size() < 10) { assistants.add(assistant); } else {
System.out.println(""Maximum number of assistants reached."");
System.exit(0); } }   public Assistant(String firstName, String lastName,
String address, String room, String phoneNumber, Professor supervisor) {
super(firstName, lastName, address, room, phoneNumber); this.supervisor =
supervisor; this.isEmployed = true; // Assistants are employed when
created if (supervisor != null) { supervisor.addAssistant(this); } }
try (BufferedWriter bw = new BufferedWriter(new
FileWriter(cheapFileName))) { for (Product product : products) { total +=
product.getPrice(); count++; minPrice = Math.min(minPrice,
product.getPrice()); maxPrice = Math.max(maxPrice, product.getPrice());
if (product.getPrice() < 20) { bw.write(product.toString());
bw.newLine(); } } } catch (IOException e) { e.printStackTrace(); }   Path
inputPath = Paths.get(""Productlist.txt""); Path modOutputPath =
Paths.get(""ModProductlist.txt""); Path cheapOutputPath =
Paths.get(""CheapProducts.txt"")  List<String> lines =
```

Files.readAllLines(inputPath); for (String line : lines) {
products.add(new Product(line)); }   Each line identifies its own
product. The attributes of a product are artno, name and price The order
of the attributes in the file is not fixed, also a artno of a product can
occur several times.   solved by for (String part : parts) { if
(part.startsWith(""name:"")) { name =
part.split("":"")[1].trim().replace(""\"""", """"); } else if
(part.startsWith(""artno:"")) { artno =
part.split("":"")[1].trim().replace(""\"""", """"); } else if
(part.startsWith(""price:"")) { price =
Integer.parseInt(part.split("":"")[1].trim()); } minPrice =
Math.min(minPrice, product.getPrice()); maxPrice = Math.max(maxPrice,
product.getPrice()); public class OuterClass { private String message =
""Hello from Outer Class""; // Inner Class class InnerClass { void
displayMessage() { System.out.println(""Message from Outer Class: "" +
message); } } public static void main(String[] args) { // Create an
instance of the Outer Class OuterClass outer = new OuterClass(); //
Create an instance of the Inner Class OuterClass.InnerClass inner =
outer.new InnerClass();   inner.displayMessage(); } }"
f no sample is tainted, the experiment was successful and test() returns
true as output.   code for (int i = 0; i < samples.length; i++) { if
(samples[i] != null && samples[i].tainted) { // If the sample is
contaminated, mark all samples as tainted for (int j = 0; j <
samples.length; j++) { if (samples[j] != null) { samples[j].tainted =
true; } } allClean = false; // The experiment is not successful break; }
} // If all samples are clean, the test was successful return allClean;
} breaking after first record is tainted , not going tocheck another ,
very imp when to use break each sample clas has 5 fastsample in this
case FastSampleSet fastSampleSet = new FastSampleSet(5) String[] s;
//not static since each object will learn new words //count for each
object means each person int count; public PersonA(String first,String
last) { super(first,last); s=new String[100]; count=0; } for(char
c1:s.toCharArray()) { c[c1]++; } for(char c3:s2.toCharArray()) {
c2[c3]++; } for(int i=0;i<256;i++) { if(c[i]!=c2[i]) { flag=false; } }
int[] charCounts = new int[26]; charCounts[s1.charAt(i) - 'a']++; , is
good way to store character in array of 26 // Frequency array for 26
lowercase English letters int[] charCounts = new int[26]; // Increment
counts for characters in s1 and decrement for characters in s2 for (int i
= 0; i < s1.length(); i++) { charCounts[s1.charAt(i) - 'a']++;
charCounts[s2.charAt(i) - 'a']--; } // Check if all counts are zero for
(int count : charCounts) { if (count != 0) { return false; } }
return true; --goo way to check if 2 string are anagrams , incrementing
and then decrementing array public static String[] checkanagrams(String
s,int parts) { //rajansah parts is 2 int n=s.length()/parts; String[]
s1=new String[parts]; for(int i=0;i<parts;i++) { //
System.out.println(s.substring(i*n, i + n)); //(1*4,4+4) s1[i] =
s.substring(i*n, i*n + n); } return s1; }
Write another class Mia that internally stores two Dice objects --means
create 2 dice object inside mia constructor
public Mia() { d1 = new Dice(); d2 = new Dice(); }
public class Mia { private Dice d1, d2; }
private static String adminPw = null; --when variable like variable is
not set , then initialted with null

// Check if user already exists for (int i = 0; i < 100; i++) { if (name.equals(id[i])) { return false; // User already exists } }
###checking user can't be done together with adding user , since we need to check first if user exist or not and then initialize with new user , for that we need 2 diff loop
Node(int data) { this.data = data; this.next = null; } initilizing every node object with null
in queue , initilizaing front,rear public Queue() { this.front = this.rear = null; this.size = 0; }
in queue , if asked about returning elem at specific pos , run a loop till one place before so that once come out of loop pointer will point to that pos elem for (int i = 0; i < position; i++) { current = current.next; }
Node current = q.front;-- imp assign for working in queue, a moving pointer which start from front