You can view this report online at : https://www.hackerrank.com/x/tests/1049020/candidates/35664131/report

Full Name:	Daniel Velasquez Iturrate
Email:	daniel.velasqueziturrate@cognizant.com
Test Name:	**CDE - Custom assessment - Problem Solving, Java, Microservices & SpringBoot - GenC**
Taken On:	31 Jan 2022 14:33:43 IST
Time Taken:	155 min 2 sec/ 160 min
Work Experience:	1 years
Invited by:	Seshadri
Invited on:	31 Jan 2022 14:31:56 IST
Skills Score:	Java (Basic) 0/75 Problem Solving (Basic) 0/75
Tags Score:	Algorithms 0/75 Implementation 0/75 Interviewer Guidelines 0/75 Java 0/75 Medium 62.91/240 Microservices 12.5/20 Problem Solving 0/75 Spring Boot 30/50 comparator 3.33/5 lambda expressions 13.33/15 streams 3.75/5

25.2%

62/250

scored in **CDE - Custom assessment - Problem Solving, Java, Microservices & SpringBoot - GenC** in 155 min 2 sec on 31 Jan 2022 14:33:43 IST

Recruiter/Team Comments:

No Comments.

Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review.

	Question Description	Time Taken	Score	Status
Q1	Prison Break > Coding	49 min 10 sec	0/ 75	!
Q2	Sport Inheritance > Coding	53 min 49 sec	0/ 75	!
Q3	Lambda Expressions > Multiple Choice	4 min 1 sec	5/ 5	✓

Q4	Streams > Multiple Choice	3 min 58 sec	3.75/ 5	✓
Q5	Lambda Expressions > Multiple Choice	2 min 1 sec	3.33/ 5	✓
Q6	Lambda Expressions > Multiple Choice	4 min 27 sec	5/ 5	✓
Q7	Comparator > Multiple Choice	5 min 5 sec	3.33/ 5	✓
Q8	Microservice Architecture > Multiple Choice	25 sec	0/ 5	✗
Q9	Microservices > Multiple Choice	44 sec	2.5/ 5	✓
Q10	Microservices > Multiple Choice	3 min 59 sec	5/ 5	✓
Q11	Microservices > Multiple Choice	3 min 31 sec	5/ 5	✓
Q12	Spring Boot - Microservices > Multiple Choice	1 min 38 sec	0/ 5	✗
Q13	Spring Boot > Multiple Choice	2 min 57 sec	0/ 5	✗
Q14	Spring Boot > Multiple Choice	1 min 27 sec	5/ 5	✓
Q15	Spring Boot > Multiple Choice	5 min 2 sec	0/ 5	✗
Q16	Spring Boot > Multiple Choice	1 min 11 sec	0/ 5	✗
Q17	Spring Boot > Multiple Choice	58 sec	5/ 5	✓
Q18	Spring Boot > Multiple Choice	1 min 15 sec	0/ 5	✗
Q19	Spring Boot > Multiple Choice	2 min	5/ 5	✓
Q20	Spring Boot > Multiple Choice	1 min 32 sec	5/ 5	✓
Q21	Spring Boot > Multiple Choice	2 min 55 sec	5/ 5	✓
Q22	Spring Boot > Multiple Choice	2 min 45 sec	5/ 5	✓

QUESTION 1

Needs Review

Score 0

Prison Break > Coding

Implementation | Medium | Problem Solving | Algorithms | Interviewer Guidelines

QUESTION DESCRIPTION

A prisoner is planning to escape from prison! The prison's gate consists of horizontal and vertical bars that are spaced one unit apart, so the area of each hole between the bars is *1 × 1*. The prisoner manages to remove certain bars to make some bigger holes. Determine the area of the largest hole in the gate after the bars are removed.

Example

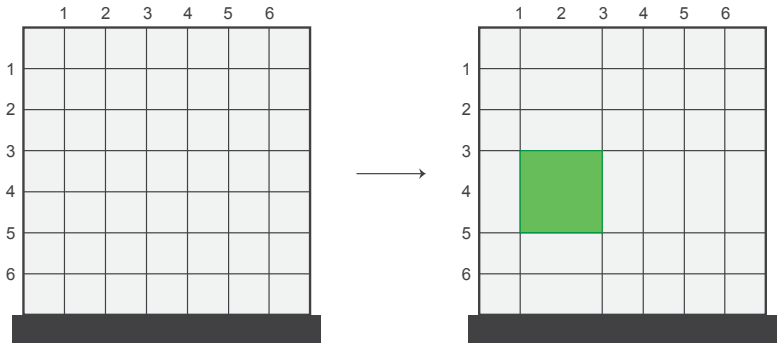
n = 6

m = 6

h = [4]

v = [2]

In the images below., the left image depicts the initial prison gate with *n = 6* horizontal and *m = 6* vertical bars. The area of the biggest hole in the bars is *1 × 1*. The right image depicts that same gate after the prisoner removes horizontal bar *4* and vertical bar *2*, at which point the area of the biggest hole in the bars becomes *2 × 2 = 4*:



Function Description

Complete the function *prison* in the editor below.

prison has the following parameter(s):

int n: integer, the number of horizontal bars initially

int m: integer, the number of vertical bars initially

int h[x]: an array of integers, the horizontal bars to remove

int v[y]: an array of integers, the vertical bars to remove

Returns:

int: a long integer denoting the *area* of the biggest hole in the prison gate's bars.

Constraints

- $1 \le n, m \le 10^5$
- $0 < x \le n$
- $0 < y \le m$
- $1 \le h[i] \le n$, where $1 \le i \le n$.
- $1 \le v[j] \le m$, where $1 \le j \le m$.
- The values in array *h* are distinct.
- The values in array *v* are distinct.

▼ Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer *n*.
The second line contains an integer *m*.
The third line contains an integer *x*, the size of the array *h*.
Each of the next *x* lines contains an integer *h[i]* where $0 \le i < n$.
The next line contains an integer *y*, the size of the array *v*.
Each of the next *v* lines contains an integer *v[j]* where $0 \le j < m$.

▼ Sample Case 0

Sample Input 0

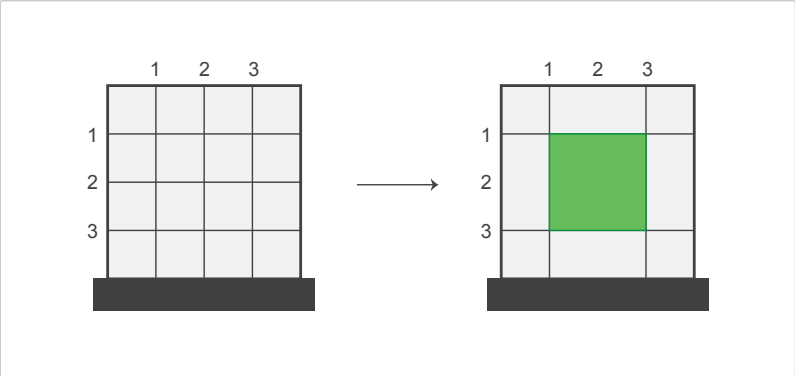
```
STDIN    Function
-----    --------
3    →   n = 3
3    →   m = 3
1    →   h[] size x = 1
2    →   h = [2]
1    →   v[] size y = 1
2    →   v = [2]
```

Sample Output 0

```
4
```

Explanation 0

There are *n = m = 3* bars in the vertical and horizontal directions. Bars to remove are *h = [2]* and *v = [2]* so the gate looks like this:



Return the area of the biggest hole, *4*, as the answer.

▼ Sample Case 1

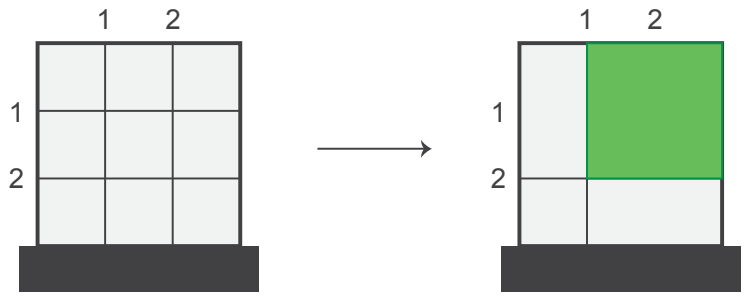
Sample Input 1

```
STDIN    Function
-----    --------
2    →   n = 2
2    →   m = 2
1    →   h[] size x = 1
1    →   h = [1]
1    →   v[] size y = 1
2    →   v = [2]
```

Sample Output 1

```
4
```

Explanation 1



There are 2 vertical and two horizontal bars initially. After removing the two bars, *h = [1]* and *v = [2]*, the top-right cell will be the largest hole with area 4.

▼ Sample Case 2

Sample Input 2

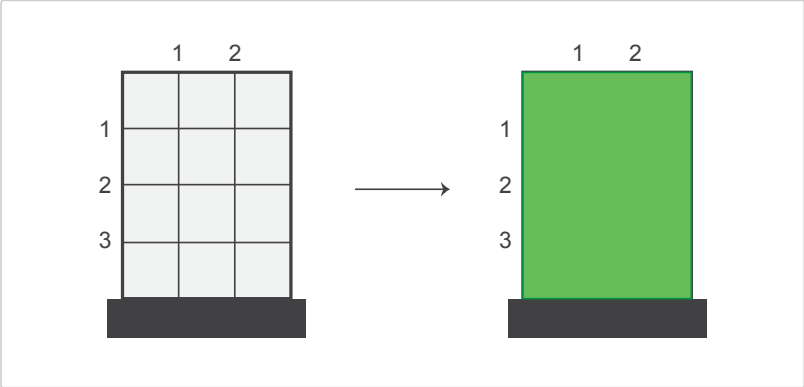
```
STDIN    Function
-----    --------
3    →   n = 3
2    →   m = 2
3    →   h[] size x = 3
1    →   h = [1, 2, 3]
2
3
2    →   v[] size y = 2
1    →   v = [1, 2]
2
```

Sample Output 2

```
12
```

Explanation 2

Initially there are *n = 3* horizontal and *m = 2* vertical bars. Remove bars *h = [1, 2, 3]* and *v = [1, 2]* so the gate looks like this:



The area of the biggest hole is *12*.

INTERVIEWER GUIDELINES

▼ Hint 1

Since there is not enough memory to create a 2-D grid of dimensions *h* x *v*, deal with the horizontal and vertical bar information separately.

▼ Hint 2

Find the maximum horizontal and vertical gaps.

▼ Solution

Concepts covered:

Arrays, Ad-Hoc

Optimal Solution:

The area of the largest hole will be the rectangular gap formed by the maximum horizontal and vertical gaps. So we take two arrays in which all elements are marked 1 initially. Here, 1 indicates that the bar is not removed. Then for each bar ID that got removed, we update the corresponding array element with 0. Here, 0 indicates that the bar got removed. Now we find the longest subarray in each array where all elements are 0. The product of the two lengths is our answer.

The time complexity of the above approach is O(N+M), where N and M are the number of horizontal and vertical bars respectively.

```
def prison(n, m, h, v):
    max_h = 1
    max_v = 1
```

```
    # initialize arrays with all bars present (1)
    prison_h = [1 for i in range(0, n+1)]
    prison_v = [1 for i in range(0, m+1)]

    # remove horizontal and vertical bars specified
    for x in h:
        prison_h[x-1] = 0

    for x in v:
        prison_v[x-1] = 0

    # find the maximum horizontal and vertical gaps
    cnt = 1
    for x in prison_h:
        if x == 0:
            cnt += 1
        else:
            max_h = max(max_h, cnt)
            cnt = 1

    cnt = 1
    for x in prison_v:
        if x == 0:
            cnt += 1
        else:
            max_v = max(max_v, cnt)
            cnt = 1

    return max_h*max_v
```

Brute Force Approach:

Create a 2-D array of size NxM, where N and M are the numbers of horizontal and vertical bars respectively, then find the largest rectangular grid where all elements are 0.

The time complexity of this approach is $O((N*M)^2)$.

Error Handling:

1. The final answer might cross the maximum integer limit in some programming languages.

▼ Complexity Analysis

Time Complexity - O(N+M).

The time complexity is O(N+M), where N and M are the number of horizontal and vertical bars respectively.

Space Complexity - O(N+M).

Since we are creating arrays of length N and M to store the bar information, the space complexity is also O(N+M), where N and M are the number of horizontal and vertical bars respectively.

CANDIDATE ANSWER

Language used: **Java 8**

```
1  import java.io.*;
2  import java.math.*;
3  import java.security.*;
4  import java.text.*;
5  import java.util.*;
6  import java.util.concurrent.*;
7  import java.util.function.*;
8  import java.util.regex.*;
9  import java.util.stream.*;
10 import static java.util.stream.Collectors.joining;
11 import static java.util.stream.Collectors.toList;
```

```
import static java.util.stream.Collectors.toList;

class Result {

    /*
     * Complete the 'prison' function below.
     *
     * The function is expected to return a LONG_INTEGER.
     * The function accepts following parameters:
     *  1. INTEGER n
     *  2. INTEGER m
     *  3. INTEGER_ARRAY h
     *  4. INTEGER_ARRAY v
     */

    public static long prison(int n, int m, List<Integer> h, List<Integer> v)
{
    boolean[] hBol = new boolean[n+1];
    Arrays.fill(hBol, true);
    boolean[] vBol = new boolean[m+1];
    Arrays.fill(vBol, true);

    for(int horizontal:h){
        hBol[horizontal] = false;
    }

    for(int vertical:v){
        vBol[vertical] = false;
    }

    int ch = 0, hMax = Integer.MIN_VALUE, cv = 0, vMax = Integer.MIN_VALUE;

    for(int i = 0; i< hBol.length; i++){
        if(hBol[i]){
            ch = 0;
        }else{
            ch++;
            hMax = Math.max(ch, hMax);
        }
    }

        for(int i = 0; i< vBol.length; i++){
        if(vBol[i]){
            cv = 0;
        }else{
            cv++;
            vMax = Math.max(cv, vMax);
        }
    }

    return (hMax+1) * (vMax+1);
}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));

        int n = Integer.parseInt(bufferedReader.readLine().trim());

        int m = Integer.parseInt(bufferedReader.readLine().trim());

        int hCount = Integer.parseInt(bufferedReader.readLine().trim());
```

```
        int hCount = Integer.parseInt(bufferedReader.readLine().trim());

        List<Integer> h = IntStream.range(0, hCount).mapToObj(i -> {
            try {
                return bufferedReader.readLine().replaceAll("\\s+$", "");
            } catch (IOException ex) {
                throw new RuntimeException(ex);
            }
        })
            .map(String::trim)
            .map(Integer::parseInt)
            .collect(toList());

        int vCount = Integer.parseInt(bufferedReader.readLine().trim());

        List<Integer> v = IntStream.range(0, vCount).mapToObj(i -> {
            try {
                return bufferedReader.readLine().replaceAll("\\s+$", "");
            } catch (IOException ex) {
                throw new RuntimeException(ex);
            }
        })
            .map(String::trim)
            .map(Integer::parseInt)
            .collect(toList());

        long result = Result.prison(n, m, h, v);

        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();

        bufferedReader.close();
        bufferedWriter.close();
    }
}
```

Result: Compilation Failed

Compile Message

```
Solution.java:106: error: reached end of file while
parsing
}
 ^
1 error
```

No Comments

QUESTION 2

Needs Review

Score 0

Sport Inheritance › Coding Medium Java

QUESTION DESCRIPTION

Create two classes: *Cricket* and *Football*. Both classes should implement the interface *Sport*.

1. Class *Cricket* should have the variable *int[ ] playerIDs*. The 1-based index of a player is the player's ID. It should have the following methods:

- *Cricket( )*: Initialize the empty array *playerIDs* to store 11 integer values. Assign the value *1* to each element in *playerIDs*. Print "*A new cricket team has been formed*".

- *void calculateAvgAge(int[] age)*: *age* contains the age of every player. Print the average age of the team with two digits after the decimal in the form "*The average age of the team is {avgAge}*".
- *void retirePlayer(int id)*: Assign -1 to *playerIds[id].* Print "*Player with id: {id} has retired*". If the player has already retired, print "*Player has already retired*".

2. Class *Football* should have the variable *int[] playerIDs.* The 1-based index of a player is the player's ID. It should have the following methods:

- *Football( )*: Initialize the empty array *playerIDs* to store 11 integer values. Assign the value *1* to each element in *playerIDs.* Print "*A new football team has been formed*".
- *void calculateAvgAge(int[] age)*: *age* contains the age of every player. Print the average age of the team with two digits after the decimal in the form "*The average age of the team is {avgAge}*".
- *void retirePlayer(int id)*: Assign -1 to *playerIDs[id].* Print "*Player with id: {id} has retired*". If the player has already retired, print "*Player has already retired*".
- *void playerTransfer(int fee, int id):* If the player is still on the team, *playerIDs[id] = 1,* print "*Player with id: {id} has been transferred with a fee of {fee}*". If the player has already retired, print "*Player has already retired*".

The locked code stub provides the interface *Sport* and also validates the implementation of the *Cricket* and *Football* classes. Please use inheritance and encapsulation to minimize code repetition.

Constraints

- $20 \leq age \leq 40$

▼ Input Format For Custom Testing

The first line contains 11 space-separated integers, the ages of the players on the cricket team.
The next line contains 11 space-separated integers, the ages of the players on the football team.
The next 5 lines each contain an integer, the id of a player to retire.
The last line contains two space-separated integers, the transfer fee and ID of a player.

▼ Sample Case 0

Sample Input For Custom Testing

```
26 32 36 33 24 31 30 35 36 21 28
32 21 27 35 24 36 31 25 23 30 22
11
7
6
1
1
659 5
```

Sample Output

```
A new cricket team has been formed
A new football team has been formed
The average age of the team is 30.18
The average age of the team is 27.82
Player with id: 11 has retired
Player with id: 7 has retired
Player with id: 6 has retired
Player with id: 1 has retired
Player has already retired
Player with id: 5 has been transferred with a fee of 659
```

INTERVIEWER GUIDELINES

```
class Cricket implements Sport{
 final int NO_OF_PLAYERS = 11;
 int[] playerIDs;

 Cricket(){
```

```
  playerIDs = new int[NO_OF_PLAYERS + 1];
  Arrays.fill(this.playerIDs, 1);
  this.playerIDs[0] = -1;
  System.out.println("A new cricket team has been formed");
 }

 public void calculateAvgAge(int[] ages){
  float avg, sum = 0;
  for(int i : ages)
   sum += i;

  System.out.format("The average age of the team is %.2f\n",sum/11f);
 }

 public void retirePlayer(int playerID){
  if(this.playerIDs[playerID] == 1){
  System.out.println("Player with id: "+playerID+" has retired");
  this.playerIDs[playerID] = -1;
  }
  else System.out.println("Player has already retired");
 }


}

class Football implements Sport{
 final int NO_OF_PLAYERS = 11;
 int[] playerIDs;

 Football(){
  playerIDs = new int[NO_OF_PLAYERS + 1];
  Arrays.fill(playerIDs,1);
  this.playerIDs[0] = -1;
  System.out.println("A new football team has been formed");
 }

 public void calculateAvgAge(int[] ages){
        float avg, sum = 0;
  for(int i : ages)
   sum += i;

  System.out.format("The average age of the team is %.2f\n",sum/11f);
 }

 public void retirePlayer(int playerID){
  if(this.playerIDs[playerID] == 1){
  System.out.println("Player with id: "+playerID+" has retired");
  this.playerIDs[playerID] = -1;
  }
  else System.out.println("Player has already retired");
 }

 public void playerTransfer(int fee,int id){
  if(this.playerIDs[id] == 1){
  System.out.println("Player with id: "+id+" has been transferred with a
fee of "+fee);
  this.playerIDs[id] = -1;
  }
  else System.out.println("The player has already retired");
 }
}
```

CANDIDATE ANSWER

Language used: **Java 8**

```
class Cricket implements Sport{
```

```
class Cricket implements Sport{
    public int[] playerIDs;
    public float sum;
    public float avgAge;

    public Cricket(){
        playerIDs = new int[11];
        playerIDs[0] = 1;
        playerIDs[1] = 1;
        playerIDs[2] = 1;
        playerIDs[3] = 1;
        playerIDs[4] = 1;
        playerIDs[5] = 1;
        playerIDs[6] = 1;
        playerIDs[7] = 1;
        playerIDs[8] = 1;
        playerIDs[9] = 1;
        playerIDs[10] = 1;
        System.out.println("A new cricket team has been formed");
    }

    public void calculateAvgAge(int[] age){
        sum = 0;
        avgAge = 0;
        for(int i = 0; i < age.length; i++){
            sum += age[i];
        }
        avgAge = sum / 11;

        System.out.println("The average of the team is " +avgAge);
    }

    public void retirePlayer(int id){
        if(playerIDs[id] == -1){
            System.out.println("Player has already retired");
        }else{
            playerIDs[id] = -1;
            System.out.println("Player with id: " +id+ " has retired");
        }
    }

}

class Football implements Sport{
    public int[] playerIDs = new int[11];
    public float sum;
    public float avgAge;

    public Football(){
        playerIDs = new int[11];
        playerIDs[0] = 1;
        playerIDs[1] = 1;
        playerIDs[2] = 1;
        playerIDs[3] = 1;
        playerIDs[4] = 1;
        playerIDs[5] = 1;
        playerIDs[6] = 1;
        playerIDs[7] = 1;
        playerIDs[8] = 1;
        playerIDs[9] = 1;
        playerIDs[10] = 1;
        System.out.println("A new footbal team has been formed");
    }

```

```
    public void calculateAvgAge(int[] age){
        sum = 0;
        avgAge = 0;
        for(int i = 0; i < age.length; i++){
            sum += age[i];
        }
        avgAge = sum / 11;

        System.out.println("The average of the team is " +avgAge);
    }

    public void retirePlayer(int id){
        if(playerIDs[id] == -1){
            System.out.println("Player has already retired");
        }else{
            playerIDs[id] = -1;
            System.out.println("Player with id: " +id+ " has retired");
        }
    }

    public void playerTransfer(int fee, int id){
        if(playerIDs[id] == 1){
            System.out.println("Player with id: " +id+ " has been transferred
with a fee of: " +fee);
        }else{
            System.out.println("Player has already retired");
        }
    }

}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Easy	Sample case	Runtime Error	0	0.1066 sec	25.1 KB
TestCase 1	Easy	Sample case	Wrong Answer	0	0.1139 sec	24.9 KB
TestCase 2	Easy	Sample case	Wrong Answer	0	0.0957 sec	25.1 KB
TestCase 3	Easy	Hidden case	Wrong Answer	0	0.1355 sec	25 KB
TestCase 4	Easy	Hidden case	Wrong Answer	0	0.1118 sec	25.2 KB
TestCase 5	Easy	Hidden case	Runtime Error	0	0.1108 sec	25.1 KB
TestCase 6	Easy	Hidden case	Runtime Error	0	0.1324 sec	25 KB
TestCase 7	Medium	Hidden case	Runtime Error	0	0.0959 sec	25 KB
TestCase 8	Medium	Hidden case	Runtime Error	0	0.1423 sec	25.2 KB
TestCase 9	Medium	Hidden case	Runtime Error	0	0.1432 sec	25 KB
TestCase 10	Medium	Hidden case	Wrong Answer	0	0.1319 sec	24.9 KB
TestCase 11	Medium	Hidden case	Wrong Answer	0	0.1106 sec	25.1 KB
TestCase 12	Medium	Hidden case	Runtime Error	0	0.1045 sec	25.1 KB

No Comments

QUESTION 3

Correct Answer

Score 5

Lambda Expressions › Multiple Choice

lambda expressions | Medium

QUESTION DESCRIPTION

```
Line 1: import java.util.*;
Line 2: import java.util.function.*;
Line 3: public class ConsumerDemo {
Line 4:     public static void main(String args[]){
Line 5:     ...
Line 6:      List<Integer> integerList=Arrays.asList(100,200,300,400);
Line 7:      printList(integerList, consumer);
Line 8:   }
Line 9:    public static void printList(List<Integer> listOfIntegers,
Consumer<Integer> consumer){
Line 10:    for(Integer integer:listOfIntegers){
Line 11:      consumer.accept(integer);
Line 12:   }
Line 13:  }
Line 14: }
```

The main() method of ConsumerDemo class is expected to print all the values in the integerList. Examine the options given below and select the statement that can be inserted at Line 5 to get the desired output?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ✓ Consumer<Integer> consumer= number -> System.out.print(number +" ");
- Consumer<Integer> consumer = number -> number;
- Consumer<Integer> consumer = number -> System.out::print;
- Consumer<Integer> consumer= consumer.andThen(number ->System.out.println(number + " "));
- Consumer<Integer> consumer = number -> value;

No Comments

QUESTION 4

Correct Answer

Score 3.75

Streams › Multiple Choice streams Medium

QUESTION DESCRIPTION

```
Take a look at the code given.
Line 1: import java.util.Arrays;
Line 2: import java.util.List;
Line 3: import java.util.stream.*;
Line 4: class SummingClass {
Line 5: public static void main(String args[]){
Line 6:  List<Integer> numbers = Arrays.asList(10,20,30,40);
Line 7:  ...
Line 8:  System.out.println(result);
Line 9:  }
Line 10: }

In the above given SummingClass, what possible statement(s) can be
inserted at Line 7 to sum all the integers given in the List 'numbers'.
Choose FOUR options.
```

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- int result = numbers.stream().reduce(0, (subtotal)) -> subtotal + element;
- ✓ int result = numbers.stream().collect(Collectors.summingInt(Integer::intValue));
- ✓ int result = numbers.parallelStream().reduce(0, Integer::sum);
- ✓ int result = numbers.stream().reduce(0, Integer::sum);
- ✓ int result = numbers.stream().reduce(0, (subtotal, element) -> subtotal + element);

No Comments

QUESTION 5

Correct Answer

Score 3.33

Lambda Expressions › Multiple Choice

lambda expressions | Medium

QUESTION DESCRIPTION

The Sorting class given below sorts the given string array based on the length of the string.

```
Line 1: import java.util.Arrays;
Line 2: import java.util.Comparator;
Line 3: public class Sorting {
Line 4: public static void main(String args[]){
Line 5:  String array[]={"SureshKumar","Ramesh", "Raj"};
Line 6:  Arrays.sort(array, new Comparator<String>(){
Line 7:   public int compare (String str1, String str2){
Line 8:    return(str1.length()-str2.length());
Line 9:   }
Line 10:  });
Line 11: Arrays.stream(array).forEach(str-> System.out.println(str));
Line 12: }
Line 13: }
```

Select the statement(s) from the below options which can replace Line 6 to Line 10 and give the same result. Choose THREE options.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ✓ Arrays.sort(array,(String str1, String str2)->{return(str1.length()-str2.length());});
- ✓ Arrays.sort(array,(str1,str2)->{return(str1.length()- str2.length());});
- ✓ Arrays.sort(array,(str1,str2)->{int ret = str1.length()-str2.length();return ret;});
- Arrays.sort(array,(str1,str2))->{return(str1.length()- str2.length());};

No Comments

QUESTION 6

Correct Answer

Score 5

Lambda Expressions › Multiple Choice

lambda expressions | Medium

QUESTION DESCRIPTION

```
Take a look at the code given.
Line 1: ...
Line 2: interface Incrementor {
Line 3: ...
Line 4: }
Line 5: class Demo {
Line 6:   public static void main(String args[]) {
Line 7:     Inrementor incrementor = (number) -> number+10;
Line 8:     System.out.println(incrementor.incrementByValue(10));
Line 9:   }
Line 10: }

When the above code is compiled and run, it is expected to print the value
20 on the default output stream. What are the statement(s) that need to be
inserted at Line 1 and Line 3 to get the expected value.
```

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ✓ Line 1: @FunctionalInterface
  Line 3: public int incrementByValue(int num);
- ✓ Line 1: Can be a blank line
  Line 3: public int incrementByValue(int num);
- Line 1: @FunctionalInterface Line 3: default int incrementByValue(int number) { return number+10; }
- Line 1: Can be a blank line Line 3: int incrementByValue(int number) { return number+10; }

No Comments

QUESTION 7

Correct Answer

Score 3.33

Comparator › Multiple Choice comparator Medium

QUESTION DESCRIPTION

For the below given Account class,

```
public class Account{
  private String id;
  public void setId(String id){
     this.id=id;
   }
 public String getId(){
    return id;
  }
}
```

Analyze the given below options and select all that defines a valid comparator.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ✓ Comparator compareId=Comparator.comparing(Account::getId);
- ✓ Comparator<Account> compareId = new Comparator<Account>(){
  public int compare(Account acc1, Account acc2) {
  return acc1.getId().compareTo(acc2.getId());
  }
  };
- ✓ Comparator compareId=Comparator.comparing(account->account.getId());
- Comparator compareId=Comparator.comparing(account.getId());

No Comments

QUESTION 8

Wrong Answer

Score 0

Microservice Architecture › Multiple Choice Microservices

QUESTION DESCRIPTION

Consider a monolithic application that runs all the services together in a single IP address. So whenever there is any downtime in one service, it will create an impact and downtime to the entire application.

Hence to overcome this impact microservices was introduced such that all the services will run independently and will have an impact to the other system.

However as these microservices are running independently, it is not possible to know which services is up/down and where it is running.

Revise this design and provide a solution such that it will solve the above problem
A. Implement Service Discovery Patterns where all the services are registered to the service registry such that the IP address of all the services is known
B. Implement Service Discovery patterns where the services are registered and the ports are exposed via REST APIs
C. Implement Service Discovery patterns where the REST API will support to fetch the IP address of the services are fetched dynamically without any prior registration

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- A,C
- ✓ A,B
- B,C
- C only

No Comments

QUESTION 9

Correct Answer

Score 2.5

Microservices › Multiple Choice Medium Microservices

QUESTION DESCRIPTION

Mr. John is challenged with a discussion on how microservices are prevailing rather than monolithic architecture, while the latter can also be dockerized to avail most of the benefits of micro services. What are the value adds which put microservices on top? Choose TWO Options.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- Less chances of maintaining the fault tolerance.
- ✓ Scalability of the applications improves the performance on servers.
- Size of the application can degrade the performance and slow down the start up.
- ✓ Majorly supports the complex application development.

No Comments

QUESTION 10

Correct Answer

Score 5

Microservices › Multiple Choice Medium Microservices

QUESTION DESCRIPTION

What are the benefit of Single Service per Host model in Microservices? Select THREE options.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- [x] It avoids single point of failure. A failure on one host will only impact one service at a time.
- [x] It makes monitoring and security easier than multiple services on one host.
- [x] It is easier to scale in case we want a specific service to be scaled up.
- [ ] It efficiently utilize resources compared to Multiple Services per Host.

No Comments

QUESTION 11

Correct Answer

Score 5

Microservices › Multiple Choice Medium Microservices

QUESTION DESCRIPTION

Select the benefits of using database per service. Select TWO options.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- [x] It ensure that the services are loosely coupled. Changes to one service's database does not impact any other services.
- [x] Each service can use the type of database that is best suited to its needs.
- [ ] Implementing business transactions that span multiple services is straightforward
- [ ] Implementing queries that join data that is now in multiple databases is easier.

No Comments

QUESTION 12

Wrong Answer

Score 0

Spring Boot - Microservices › Multiple Choice

QUESTION DESCRIPTION

The train ticket management system allows the customer to book the tickets in various ways like normal booking, takal and premium takal booking. The portal also allows the customer to login and view the booked history or look for available train for various routes. Currently there is a single controller and services of Spring Boot application that takes cares of all the above activities.

As the request and response of this portal is handled by single server and client, there is a performance issue.
Revise this design by choosing one of the below options in such a way that there will be number of services for each request and the performance will be handled by load balancer without affecting the usage of the portal by the customer.

1. Create Eureka Server , which in turn treats itself as a client, where every services is registered to get a dependent microservices to get the job done.
2. Create seperate service class for each use cases of the business need and configure the same in SpringBootApplication class
3. Create multiple REST Client of Spring Boot application to handle the service class

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ✓ I only
- II only
- Both I and II
- III only

No Comments

QUESTION 13

Wrong Answer

Score 0

Spring Boot › Multiple Choice Medium Spring Boot

QUESTION DESCRIPTION

Consider the given property file (insurance.properties) having the following property details.

policy.lifeInsurance=YEARLY_PREMIUM_OFFER
policy.medicalInsurance=MASTER_HEALTH_CHECKUP_OFFER

From the given options, choose the right definition of PolicyManagement class, that reads from this property file and assigns to its member variable using Spring Boot.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- [x] @ConfigurationProperties(prefix="policy")
public class PolicyManagement {
private String lifeInsurance;
private String medicalInsurance;
//getters and Setters for the above
}

- [ ] @TypeSafeConfigurationProperty(prefix="policy")
public class PolicyManagement {
private String lifeInsurance;
private String medicalInsurance;

//getters and Setters for the above }

- [ ] @ConfigurationProperties
public class PolicyManagement {

@Value{policy.lifeInsurance}
private String lifeInsurance;

@Value{policy.medicalInsurance}
private String medicalInsurance;
}

- [ ] @ConfigurationProperties
public class PolicyManagement {

@Value{"$policy.lifeInsurance"}
private String lifeInsurance;

@Value{"$policy.medicalInsurance"}
private String medicalInsurance;
}

No Comments

QUESTION 14

Correct Answer

Score 5

Spring Boot › Multiple Choice Medium Spring Boot

QUESTION DESCRIPTION

What are all the HTTP methods that @RequestMapping can use while you want a Spring Boot class act as a Rest Controller?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- GET
- POST
- PUT
- ✓ All of the listed options.

No Comments

QUESTION 15

Wrong Answer

Score 0

Spring Boot › Multiple Choice Medium Spring Boot

QUESTION DESCRIPTION

Shopping Management System allows the customers to order products online. If a customer searches for a product that is not in stock, a custom response, 'Out of stock', along with HTTP_NOT FOUND status should be returned. This exception should be handled using the Spring Boot Exception Handler.
Which of the following code snippets implements the mentioned requirement?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- [ ] @ExceptionHandler(ShoppingManagementException.class)
public final ResponseEntity<ExceptionResponse>
invalidData(ShoppingManagementException exception) {
ExceptionResponse exceptionResponse = new
ExceptionResponse(exception.getMessage(), "OutOfStock"); return new
ResponseEntity<ExceptionResponse>(exceptionResponse, new
HttpHeaders(), HttpStatus.NOT_FOUND);
}

- [ ] @ExceptionHandler(ShoppingManagementException.class)
public final ResponseEntity<ExceptionResponse>
invalidData(ShoppingManagementException exception) {
ExceptionResponse exceptionResponse = new
ExceptionResponse(HttpStatus.NOT_FOUND), "OutOfStock"); return new
ResponseEntity<ExceptionResponse>(exceptionResponse);
}

- [x] @ExceptionHandler(ShoppingManagementException.class)
public final ResponseEntity<ExceptionResponse>
invalidData(ShoppingManagementException exception) {
ExceptionResponse exceptionResponse = new
ExceptionResponse(exception.getMessage());
return new ResponseEntity<ExceptionResponse>
(exceptionResponse,"OutOfStock", HttpStatus.NOT_FOUND); }

- [ ] @ExceptionHandler(ShoppingManagementException.class)
public final ResponseEntity<ExceptionResponse>
invalidData(ShoppingManagementException exception) {
ExceptionResponse exceptionResponse = new
ExceptionResponse(exception.getMessage(), "OutOfStock"); return new
ResponseEntity<ExceptionResponse>(new HttpHeaders(),
HttpStatus.NOT_FOUND);
}

No Comments

QUESTION 16

Wrong Answer

Score 0

Spring Boot › Multiple Choice Medium Spring Boot

QUESTION DESCRIPTION

Consider a bank management application that maintains all the service related classes in the package com.bank.service.* Choose a code snippet that will load all the beans in the package during the Spring Boot application start up.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- [x] package com.bank.config;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
@ComponentScan({
"com.bank.controller,com.bank.service,com.bank.dao" })
public class BankManagementApplication {
public static void main(String[] args) {
SpringApplication.run(BankManagementApplication.class, args);
}
}

- [ ] package com.bank.config;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
@Configuration({ "com.bank.controller,com.bank.service,com.bank.dao" })
public class BankManagementApplication {
public static void main(String[] args) {
SpringApplication.run(BankManagementApplication.class, args);
}
}

- [ ] package com.bank.config;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
@EnableAutoConfiguration
@ComponentScan({
"com.bank.controller,com.bank.service,com.bank.dao" })
public class BankManagementApplication {
public static void main(String[] args) {
SpringApplication.run(BankManagementApplication.class, args);
}
}

- [ ] package com.bank.config;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
@EnableAutoConfiguration({
"com.bank.controller,com.bank.service,com.bank.dao" })
@Configuration @ComponentScan({
"com.bank.controller,com.bank.service,com.bank.dao" })
public class BankManagementApplication {
public static void main(String[] args) {
SpringApplication.run(BankManagementApplication.class, args);
}
}

No Comments

QUESTION 17

Correct Answer

Score 5

Spring Boot › Multiple Choice Medium Spring Boot

QUESTION DESCRIPTION

If you do not want to do default component scan in Spring Boot application, how to disable it?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- @ComponentScan(disable)
- ✓ @ComponentScan(useDefaultFilters=false)
- @ComponentScan(useDefaultFilter=true)
- None of the listed options.

No Comments

QUESTION 18

Wrong Answer

Score 0

Spring Boot › Multiple Choice Medium Spring Boot

QUESTION DESCRIPTION

Assume in a given scenario, a checked exception occurred in a DAO component and the data flow is in a chain transaction. What will happen to the previously executed transaction?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- The previous transaction gets rollback even if transaction annotation is not used.
- ✓ The previous transaction do not get rollback automatically even if transaction annotation is used.
- The previous transaction gets rollback even if transaction annotation is used.
- None of the listed options.

No Comments

QUESTION 19

Correct Answer

Score 5

Spring Boot › Multiple Choice Medium Spring Boot

QUESTION DESCRIPTION

In a Spring Boot application, a microservice need to invoke another microservice in asynchronous way. Select from the given options to implement it. Choose Three options

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- [x] Need to use @EnableAsync in the spring boot application startup class.
- [x] Need to annotate with @Async in the method from where the other microservice will get invoked .
- [x] Need to have the return type of the method invoking the other microservice as CompletableFuture<T> .
- [ ] Need to use ExecuterService class in order to invoke the microservice asynchronously.

No Comments

QUESTION 20

Correct Answer

Score 5

Spring Boot › Multiple Choice Medium Spring Boot

QUESTION DESCRIPTION

Identify from the given options the file types to be used for a file to hold application specific key/value pairs. Choose TWO options

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- [x] application.properties
- [x] application.yml
- [ ] springboot.properties
- [ ] None of the listed options.

No Comments

QUESTION 21

Correct Answer

Score 5

Spring Boot › Multiple Choice Medium Spring Boot

QUESTION DESCRIPTION

Select from the given options the way to handle exception globally in Spring Boot application.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- Using @ExceptionHandler at class level.
- ✓ Using @ControllerAdvice at class level.
- Using @HandlerExceptionResolver at class level.
- None of the listed options.

No Comments

QUESTION 22

Correct Answer

Score 5

Spring Boot › Multiple Choice Medium Spring Boot

QUESTION DESCRIPTION

You do not want to restart the Spring boot app to reflect the change in configuration file.How to reload the configuration properties without restarting the app? Select your recommendations from the following options.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- Adding @RefreshConfig at the controller class.
- Adding @RefreshProperties at the controller class.
- ✓ Adding @RefreshScope at the controller class.
- None of the listed options.

No Comments