Rajalakshmi Engineering College

Name: kayalvizhi R
Email: 240701251@rajalakshmi.edu.in
Roll no: 240701251
Phone: 9944415744
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_Week 12_Java_Lamba Expressions_MCQ

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : MCQ

1. What is a lambda expression in Java?

Answer

A way to define anonymous methods

Status : Correct ***Marks : 1/1***

2. What is the syntax for a basic lambda expression in Java?

Answer

(parameters) -> expression

Status : Correct ***Marks : 1/1***

3. Which functional interface in Java takes two arguments and returns a result?

Answer

BiFunction

Status : Correct ***Marks : 1/1***

4. What is the return type of a lambda expression in Java?

Answer

The return type is inferred from the context

Status : Correct ***Marks : 1/1***

5. Which of the following is a valid lambda expression in Java?

Answer

All of the mentioned options

Status : Correct ***Marks : 1/1***

6. Can a lambda expression in Java have a body with multiple statements?

Answer

Yes, if the statements are enclosed in curly braces

Status : Correct ***Marks : 1/1***

7. Which of the following interfaces is NOT a functional interface in Java?

Answer

Iterable

Status : Correct ***Marks : 1/1***

8. Can a lambda expression in Java have a body with multiple statements?

Answer

Yes, if the statements are enclosed in curly braces

Status : Correct ***Marks : 1/1***

9. Can a lambda expression have more than one parameter?

Answer

Yes, it can have multiple parameters

Status : Correct ***Marks : 1/1***

10. Which functional interface is commonly used with lambda expressions in Java?

Answer

Runnable

Status : Correct ***Marks : 1/1***

Rajalakshmi Engineering College

Name: kayalvizhi R
Email: 240701251@rajalakshmi.edu.in
Roll no: 240701251
Phone: 9944415744
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 12_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Sabrina is working on a project that involves analyzing a set of numbers. In her exploration, she encounters scenarios where extracting even numbers and finding their sum is essential.

Create a program that calculates the sum of even numbers from a given array of integers using a lambda expression.

Input Format

The first line of input consists of an integer N, representing the size of the array.

The second line consists of N space-separated integers, representing the elements of the array.

Output Format

The output prints the sum of the even integers from the array.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
29 37 45

Output: 0

Answer

```
// You are using Java
import java.util.*;
import java.util.stream.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        int sum = Arrays.stream(arr)
                        .filter(x -> x % 2 == 0)   // Lambda expression to filter even numbers
                        .sum();

        System.out.println(sum);
        sc.close();
    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: kayalvizhi R
Email: 240701251@rajalakshmi.edu.in
Roll no: 240701251
Phone: 9944415744
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 12_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Alex is learning about Java's functional interfaces and lambda expressions.

He wants to write a simple program that prints the square of each number in an array using a predefined functional interface.

Help Alex complete this task using the Consumer functional interface.

Input Format

- The first line contains an integer N, the number of elements in the array.
- The second line contains N space-separated integers.

Output Format

- Print the squares of all elements in the array, separated by a space.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4
1 2 3 4
Output: 1 4 9 16

Answer

```
// You are using Java
import java.util.*;
import java.util.function.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        Consumer<Integer> squarePrinter = x -> System.out.print((x * x) + " ");

        for (int num : arr) {
            squarePrinter.accept(num);
        }

        sc.close();
    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: kayalvizhi R
Email: 240701251@rajalakshmi.edu.in
Roll no: 240701251
Phone: 9944415744
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 12_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In the mystical realm of programming, there exists a magical incantation to reveal hidden words.

Elara, the skilled enchantress, wishes to summon a word using her spell and then reverse its characters to uncover its enchanted reflection.

Write a program that uses the predefined functional interface Supplier<String> and a lambda expression to:

Supply (generate) a string, and

Display its reversed form.

Input Format

No input is required from the user.

The string must be supplied internally using a Supplier<String>.

Output Format

Print the reversed version of the supplied string.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Wizard!!
Output: !!draziW

Answer

```
// You are using Java
import java.util.*;
import java.util.function.Supplier;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String input = sc.nextLine(); // read the supplied string

        Supplier<String> supplier = () -> input;

        String original = supplier.get();
        String reversed = new StringBuilder(original).reverse().toString();

        System.out.println(reversed);
    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: kayalvizhi R
Email: 240701251@rajalakshmi.edu.in
Roll no: 240701251
Phone: 9944415744
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 12_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Abi is working on a text analysis project where she needs to categorize words based on their length.

Words that have three or fewer characters are considered “Short”, while

words with more than three characters are classified as “Long.”

Write a Java program that takes a sentence as input, analyzes each word, and prints a list showing whether each word is “Short” or “Long.”

Use the predefined functional interface Function<String, String> along with a lambda expression for categorization.

Input Format

A single line containing a sentence (words separated by spaces).

Output Format

- A single line with each word categorized as "Short" or "Long", separated by spaces.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: I love my cat
Output: Short Long Short Short

Answer

```
// You are using Java
import java.util.*;
import java.util.function.Function;

public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String sentence = sc.nextLine();

      Function<String, String> categorize = word -> word.length() <= 3 ? "Short" :
"Long";

      String[] words = sentence.split("\\s+");
      for (String word : words) {
         System.out.print(categorize.apply(word) + " ");
      }
   }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: kayalvizhi R
Email: 240701251@rajalakshmi.edu.in
Roll no: 240701251
Phone: 9944415744
Branch: REC
Department: CSE - Section 3
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_Week 12_Java_Lamba Expressions_PAH

Attempt : 1
Total Mark : 40
Marks Obtained : 37.5

Section 1 : COD

1. Problem Statement

Sneha is developing a feature for an e-commerce application that helps display product details after applying a seasonal discount.

She decides to use lambda expressions with the Consumer functional interface to print each product's name, original price, and discounted price neatly.

The program should:

Accept a list of product names and their prices.Apply a 15% discount on all products.Use a Consumer lambda expression to display the details in a formatted manner.

Input Format

The first line of input consists of an integer n, representing the number of products.

The next n lines each contain a String (product name) and a double (price) separated by a space.

Output Format

For each product, print the details in the format:

Product: <name>, Original Price: <price>, Discounted Price: <discounted price>

If there are no products, print:

No products available

Sample Test Case

Input: 1
Phone 60000
Output: Product: Phone, Original Price: 60000.0, Discounted Price: 51000.0

Answer

```
import java.util.*;
import java.util.function.Consumer;

public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      // Check if there is any integer input
      if (!sc.hasNextInt()) {
         System.out.println("No products available");
         return;
      }

      int n = sc.nextInt();
      sc.nextLine(); // consume newline

      if (n <= 0) {
         System.out.println("No products available");
         return;
      }
```

```
        List<String> names = new ArrayList<>();
        List<Double> prices = new ArrayList<>();

        // Read product details
        for (int i = 0; i < n; i++) {
            String line = sc.nextLine().trim();
            if (line.isEmpty()) continue;

            // Handle product names with spaces by splitting from last space
            int lastSpace = line.lastIndexOf(' ');
            String name = line.substring(0, lastSpace);
            double price = Double.parseDouble(line.substring(lastSpace + 1));

            names.add(name);
            prices.add(price);
        }

        // Consumer lambda to display product details with discount
        Consumer<Integer> display = i -> {
            double discountedPrice = prices.get(i) * 0.85; // 15% discount
            System.out.println("Product: " + names.get(i)
                    + ", Original Price: " + prices.get(i)
                    + ", Discounted Price: " + discountedPrice);
        };

        // Display all products
        for (int i = 0; i < names.size(); i++) {
            display.accept(i);
        }
    }
}
```

Status : Partially correct **Marks : 7.5/10**

2. Problem Statement

Rishi is working as an HR analyst in a software company. He wants to filter a list of employees based on their salary using modern Java techniques. He has a list of employee names and salaries and wants to use lambda expressions to filter those who earn more than a specific threshold.

Implement a program using lambda expressions and functional interfaces to print the names of employees whose salary is greater than or equal to 50,000.

Input Format

The first line of input consists of an integer n, representing the number of employees.

The next n lines. Each line contains a String (employee name) and an int (salary).

Output Format

The output prints the names of employees whose salary is greater than or equal to 50000, each on a new line.

If no employee found with salary greater than 50000, print: No employee found with salary >= 50000

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4
Amit 45000
Sneha 50000
Ravi 60000
Priya 30000

Output: Sneha
Ravi

Answer

```
import java.util.*;
import java.util.function.Predicate;

public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int n = Integer.parseInt(sc.nextLine().trim());
```

```
        List<String> names = new ArrayList<>();
        List<Integer> salaries = new ArrayList<>();

        for (int i = 0; i < n; i++) {
            String line = sc.nextLine().trim();
            String[] parts = line.split(" ");
            String name = parts[0];
            int salary = Integer.parseInt(parts[1]);
            names.add(name);
            salaries.add(salary);
        }

        // Lambda Predicate to check if salary >= 50000
        Predicate<Integer> isHighSalary = s -> s >= 50000;

        boolean found = false;
        for (int i = 0; i < n; i++) {
            if (isHighSalary.test(salaries.get(i))) {
                System.out.println(names.get(i));
                found = true;
            }
        }

        if (!found) {
            System.out.println("No employee found with salary >= 50000");
        }
    }
}
```

Status : Correct **Marks : 10/10**

3. Problem Statement

Emily, an analyst at a data processing firm, is tasked with cleaning up datasets to remove duplicate values from lists of integers.

Create a Java program that allows Emily to input a series of integers, with the program then utilizing a lambda expression to efficiently remove any duplicates.

Input Format

The first line of input consists of an integer N, representing the size of the array.

The second line consists of N space-separated integers, each denoting an array element.

Output Format

The output prints the array elements after removing the duplicates inside the square bracket separated by a comma and space.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 15
1 2 3 4 3 2 1 2 3 4 4 4 5 5 6

Output: [1, 2, 3, 4, 5, 6]

Answer

```
// You are using Java
import java.util.*;
import java.util.stream.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        List<Integer> numbers = new ArrayList<>();

        for (int i = 0; i < n; i++) {
            numbers.add(sc.nextInt());
        }

        // Using Stream + Lambda to remove duplicates
        List<Integer> uniqueNumbers = numbers.stream()
                                      .distinct()  // removes duplicates
                                      .collect(Collectors.toList());

        System.out.println(uniqueNumbers);
```

```
    }
}
```

Status : Correct ***Marks : 10/10***

4. Problem Statement

Aditya is developing a reading app that recommends books to users based on a predefined list.

Each time a user opens the app, it should supply the next book title in the list, one at a time, using a lambda expression and the Supplier functional interface.

When all books have been recommended, the list should start again from the beginning.

Input Format

The first line contains an integer n — the total number of available book titles.

The next n lines each contain a book title (a string).

The next line contains an integer m — the number of times users open the app (i.e., the number of recommendations to be made).

Output Format

Print the supplied book title for each recommendation, one per line.

If m > n, repeat the list from the start.

Sample Test Case

Input: 3
The Alchemist
Atomic Habits
Ikigai
5
Output: The Alchemist
Atomic Habits
Ikigai

The Alchemist
Atomic Habits

Answer

```
// You are using Java
import java.util.*;
import java.util.function.Supplier;

public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      int n = Integer.parseInt(sc.nextLine().trim());
      List<String> books = new ArrayList<>();

      for (int i = 0; i < n; i++) {
         books.add(sc.nextLine().trim());
      }

      int m = Integer.parseInt(sc.nextLine().trim());

      // index tracker (final array to allow modification inside lambda)
      int[] index = {0};

      // Supplier lambda to provide next book
      Supplier<String> bookSupplier = () -> {
         String book = books.get(index[0]);
         index[0] = (index[0] + 1) % books.size(); // wrap around
         return book;
      };

      // Generate m recommendations
      for (int i = 0; i < m; i++) {
         System.out.println(bookSupplier.get());
      }
   }
}
```

Status : Correct ***Marks : 10/10***