

NAME : Priyanka Kani

DEPT: B.Sc CS with CTIS

Scenario Based Questions

Topic: Java

- 1. Write a Java Program to iterate ArrayList using for-loop, while-loop, and advance forloop to get the result as shown below:**

```
3
While Loop:
20
30
40
Advanced For Loop:
20
30
40
For Loop:
20
30
40
```

SOLUTION :

```
package edu; import
java.util.*; public class
array_list {
public static void main(String[] args) {
    System.out.println("3");
    System.out.println("While Loop:");
    ArrayList<Integer> al = new ArrayList<Integer>();
    {
        al.add(20);
        al.add(30);
        al.add(40);
        int val = 0;
```

```

        while (al.size() > val) {
            System.out.println(al.get(val));
        }

        System.out.println("Advanced For Loop:");
        {
            List<Integer> numbers = Arrays.asList(20,30,40);

            for (Integer i : numbers)
                System.out.println(i + " ");

            System.out.println("For Loop:");
            {
                List<Integer> numbers = Arrays.asList(20,30,40);
                for (int i = 0; i < numbers.size(); i++)
                    System.out.println(numbers.get(i) + " ");
            }
        }
    }
}

```

OUTPUT :

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure, including the 'edu' package and its sub-packages 'src' and 'module-info.java'. The main editor window shows the 'array_list.java' file with the following code:

```

1 package edu;
2 import java.util.*;
3 public class array_list {

```

The Console window at the bottom shows the output of the program:

```

<terminated> array_list [Java Application] C:\Program Files\Java\jdk-17.0.4\bin\javaw.exe (20-Oct-2022, 3:01:15 pm - 3:01:15 pm) [pid: 8568]
3
While Loop:
20
30
40
Advanced For Loop:
20
30
40
For Loop:
20
30
40

```

2. Create a doubly linked list and rotate it by n node to get the result as shown below:

Original List:

1 2 3 4 5

Updated List:

4 5 1 2 3

SOLUTION :

```
package edu;

public class doubly_linked_list {

    static class Node
    {
        char data;
        Node prev;
        Node next;
    }
    static Node head = null;
    static void rotate( int N)
    {
        if (N == 0)
            return;
        Node current = head;
        int count = 1;
        while (count < N && current != null)
        {
            current = current.next;
            count++;
        }

        if (current == null)
            return;

        Node NthNode = current;

        while (current.next != null)
            current = current.next;

        current.next = head;

        (head).prev = current;

        head = NthNode.next;

        (head).prev = null;

        NthNode.next = null;
    }
}
```

```

        static void push(char new_data)
        {
            Node new_node = new Node();
            new_node.data = new_data;
            new_node.prev = null;
            new_node.next =
(head);
            if ((head) != null)
(head).prev = new_node;
            head =
new_node;
        }

        static void printList(Node node)
        {
            while (node != null && node.next != null)
            {
                System.out.print(node.data + " ");
                node = node.next;
            }
            if (node != null)
                System.out.print(node.data);
        }

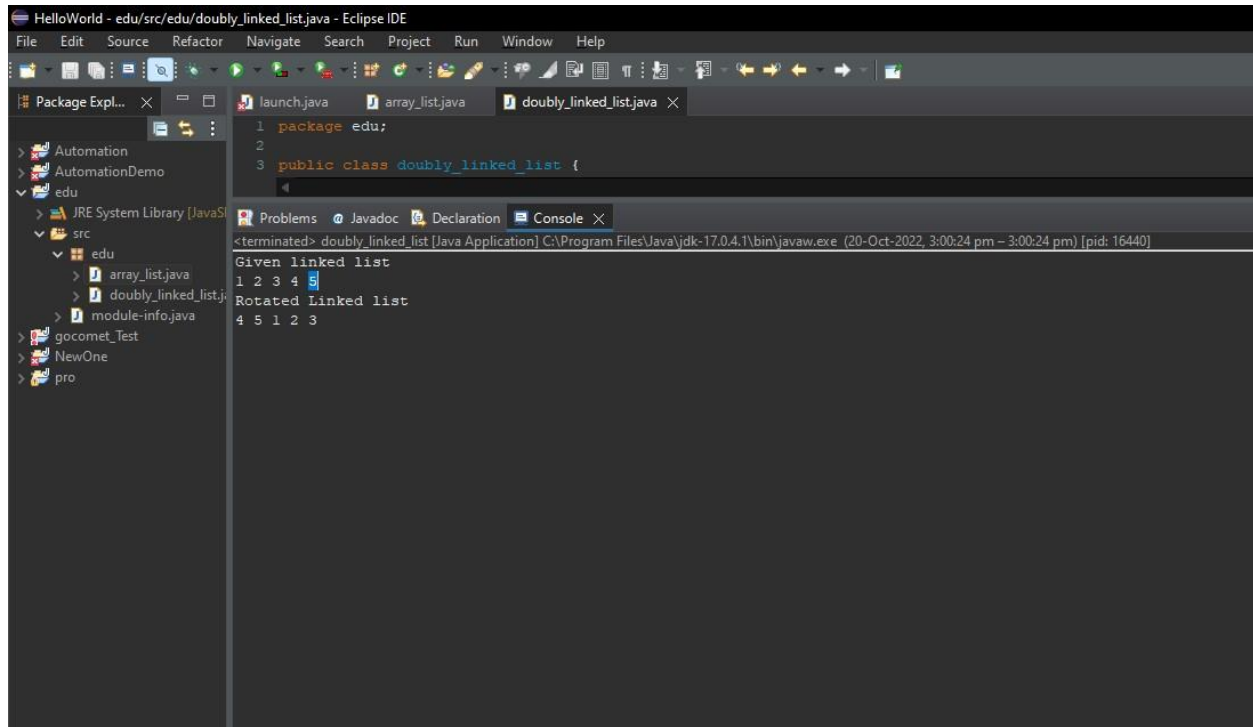
        public static void main(String[] args)
        {
            push( '5');
            push( '4');
            push('3');
            push('2');
            push( '1');

            int N = 3;

            System.out.println("Given linked list ");
            printList(head);
            rotate( N);
            System.out.println();
            System.out.println("Rotated Linked list ");
            printList(head);
        }
    }

```

OUTPUT :



```
package edu;

public class doubly_linked_list {

}

<terminated> doubly_linked_list [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (20-Oct-2022, 3:00:24 pm - 3:00:24 pm) [pid: 16440]
Given linked list
1 2 3 4
Rotated Linked list
4 5 1 2 3
```

Topic: SQL

3. At St. Xavier's College, a faculty has the following data in My SQL in database named as Class having table student related to Semester Examination

Enrollment No.	Student Name	Section	Subject Id	Marks
1	Tim	A	1	70
2	Jim	A	2	75
3	Kim	B	3	65
4	Tom	B	4	77
5	John	C	5	60
6	Joe	C	1	82
7	James	B	2	76
8	Henry	C	5	68
9	Matt	B	3	71

10	Paul	A	4	79
----	------	---	---	----

The faculty needs a section-wise Number of candidates who have secured more than or equal to 75 marks in the Semester Exam.

Note: Enrollment No. is declared as Primary Key

Output Table

Section	No. of Candidate greater than or equal to 75 marks
A	3
B	4
C	3

Can you suggest how this can be achieved? Write steps along with output screenshot

SOLUTION :

-- create

```
CREATE TABLE SEMESTER (
  Enrollment_No varchar(13) PRIMARY KEY,
  Student_Name varchar(12),
  Section varchar(7),
  Subject_id varchar(10),
  Marks int
);
```

-- insert

```
INSERT INTO SEMESTER VALUES (1, 'Tim', 'A',1,70);
INSERT INTO SEMESTER VALUES (2, 'Jim', 'A',2,75);
INSERT INTO SEMESTER VALUES (3, 'Kim', 'B',3,65);
INSERT INTO SEMESTER VALUES (4, 'Tom', 'B',4,77);
INSERT INTO SEMESTER VALUES (5, 'John','C',5,60);
INSERT INTO SEMESTER VALUES (6, 'Joe', 'C',1,82);
INSERT INTO SEMESTER VALUES (7, 'James','B',2,76);
INSERT INTO SEMESTER VALUES (8, 'Henry','C',5,68);
INSERT INTO SEMESTER VALUES (9, 'Matt','B',3,71);
```

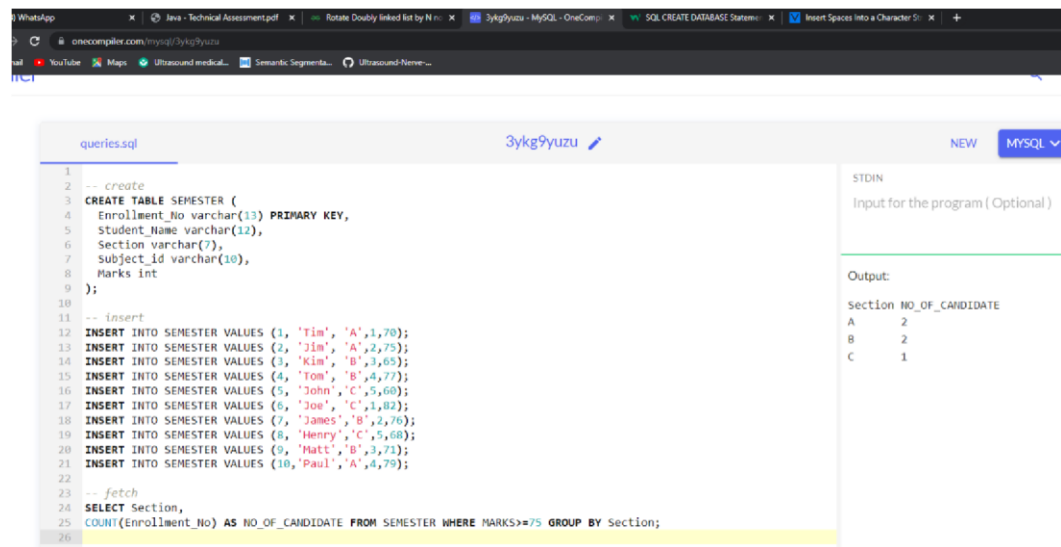
INSERT INTO SEMESTER VALUES (10,'Paul','A',4,79);

-- fetch

SELECT Section,

COUNT(Enrollment_No) AS NO_OF_CANDIDATE FROM SEMESTER WHERE MARKS>=75 GROUP BY
Section;

OUTPUT :



The screenshot shows a web browser with a SQL IDE interface. The editor contains a SQL script with two parts: a table creation and insertion, and a query. The output of the query is displayed on the right side of the interface.

```
1 -- create
2
3 CREATE TABLE SEMESTER (
4   Enrollment_No varchar(13) PRIMARY KEY,
5   Student_Name varchar(12),
6   Section varchar(7),
7   Subject_id varchar(10),
8   Marks int
9 );
10
11 -- insert
12 INSERT INTO SEMESTER VALUES (1, 'Tim', 'A', 1, 70);
13 INSERT INTO SEMESTER VALUES (2, 'Jim', 'A', 2, 75);
14 INSERT INTO SEMESTER VALUES (3, 'Kim', 'B', 3, 65);
15 INSERT INTO SEMESTER VALUES (4, 'Tom', 'B', 4, 72);
16 INSERT INTO SEMESTER VALUES (5, 'John', 'C', 5, 60);
17 INSERT INTO SEMESTER VALUES (6, 'Jow', 'C', 1, 62);
18 INSERT INTO SEMESTER VALUES (7, 'James', 'B', 2, 76);
19 INSERT INTO SEMESTER VALUES (8, 'Henry', 'C', 5, 68);
20 INSERT INTO SEMESTER VALUES (9, 'Matt', 'B', 3, 71);
21 INSERT INTO SEMESTER VALUES (10, 'Paul', 'A', 4, 79);
22
23 -- fetch
24 SELECT Section,
25 COUNT(Enrollment_No) AS NO_OF_CANDIDATE FROM SEMESTER WHERE MARKS>=75 GROUP BY Section;
```

Output:

Section	NO_OF_CANDIDATE
A	2
B	2
C	1