DAY-2:

Arithmetic Exception:

Program-1:

class ExceptionExample {

public static void main(String[] args) {

try {

int a = 10, b = 0;

int result = a / b;

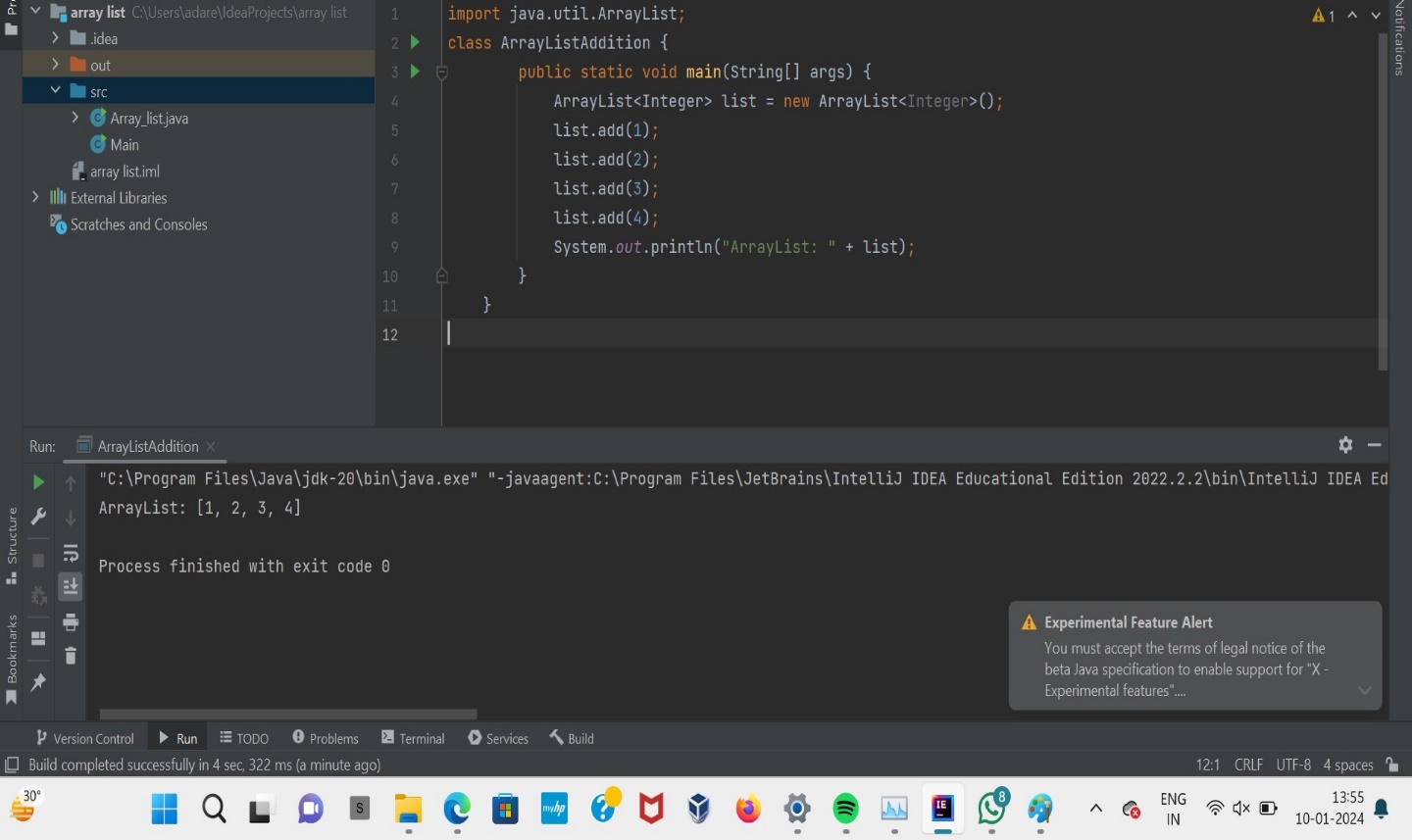
} catch (ArithmeticException e) {

System.out.println("Error: " + e.getMessage());

}

}

}

Output: 

Program:2

import java.util.ArrayList;

class ArrayListAddition {

public static void main(String[] args) {

ArrayList<Integer> list = new ArrayList<Integer>();

list.add(1);

list.add(2);

list.add(3);

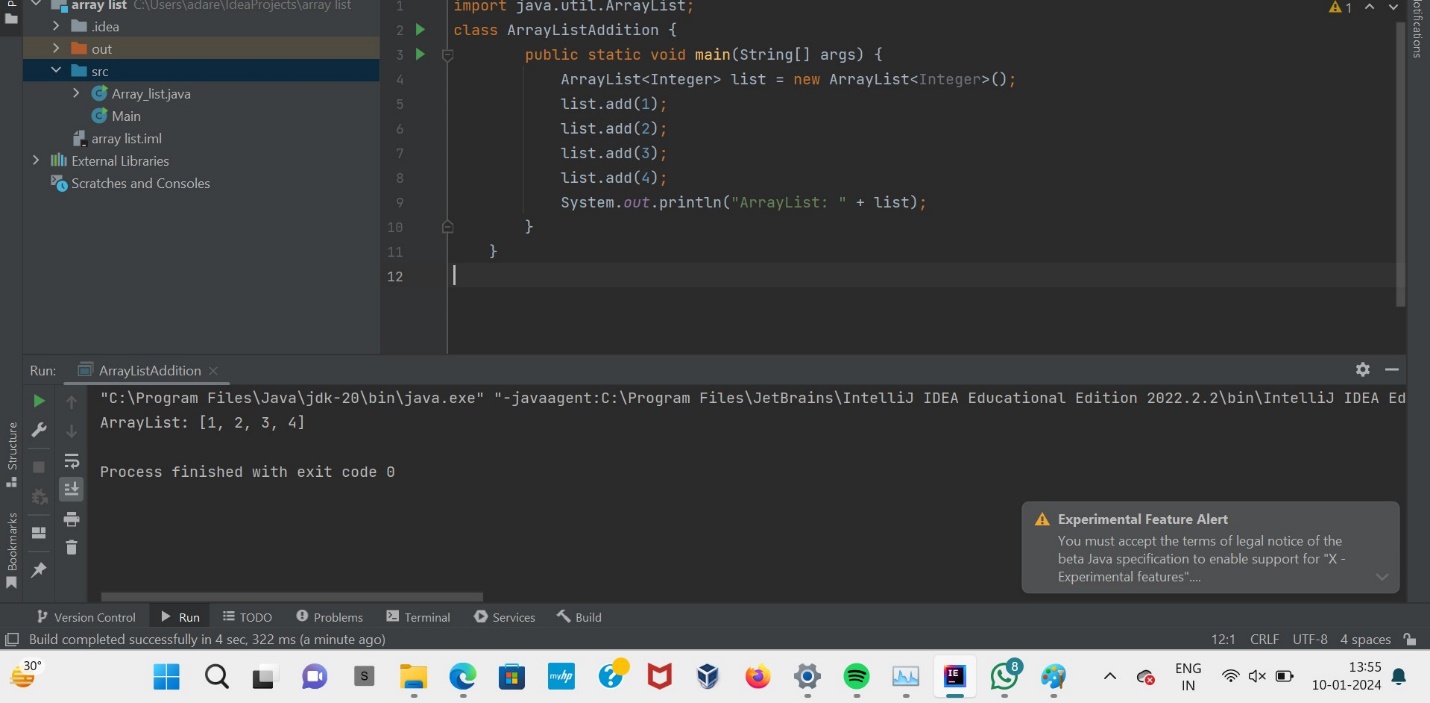
list.add(4);

System.out.println("ArrayList: " + list);

}

}

Output:



Program:3

class daddy{

void dis(int i){

System.out.println(i);

}

}

class derived extends daddy{

void dis(double i){

System.out.println(i);

}

}

class base {

public static void main(String[] args) {

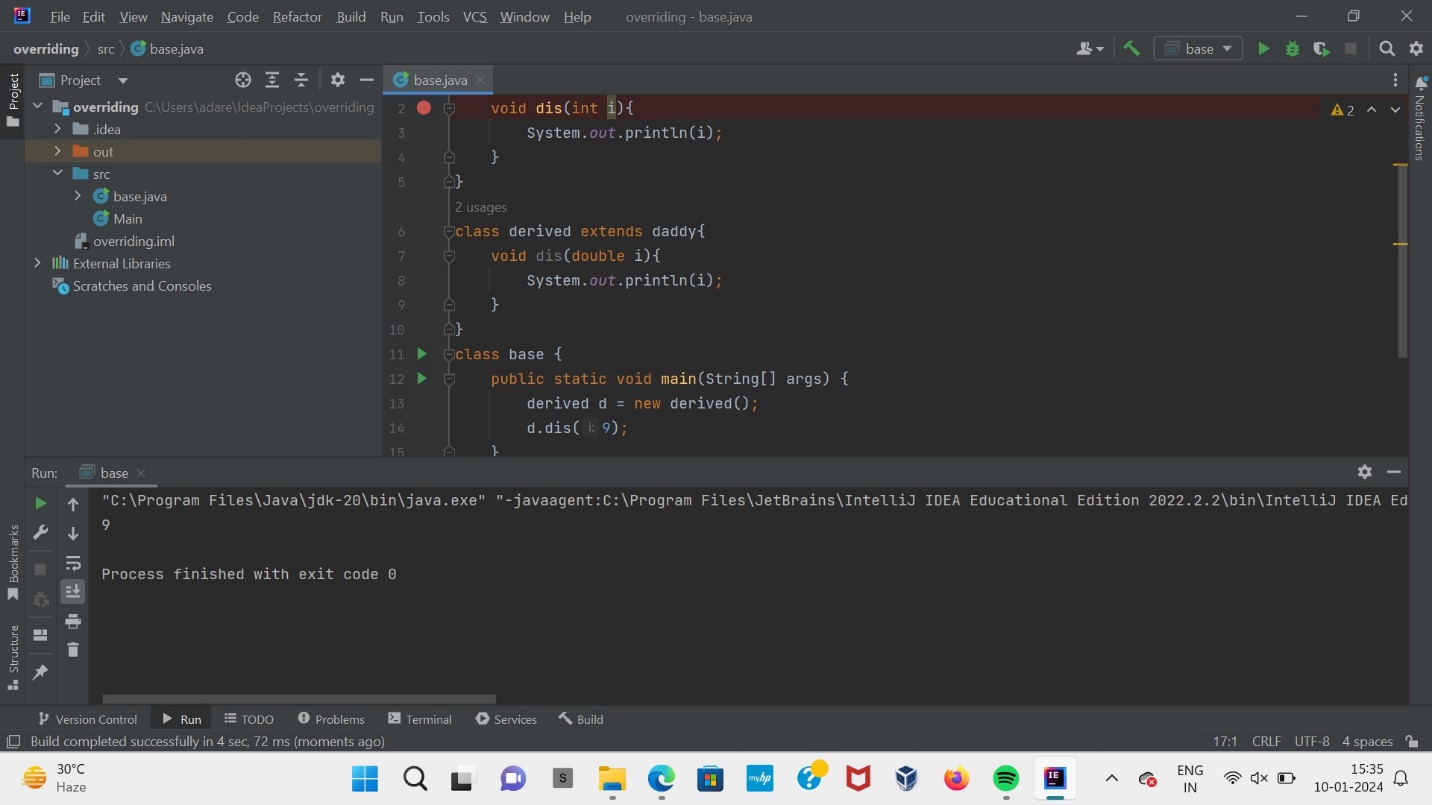
derived d = new derived();

d.dis(9);

}

}

Output:



Program:4

import java.util.Arrays;

public class ArrayRemoval {

public static void main(String[] args) {

int[] arr = {1, 2, 3, 4, 5};

int indexToRemove = 2;

for (int i = indexToRemove; i < arr.length - 1; i++) {

arr[i] = arr[i+1];

}

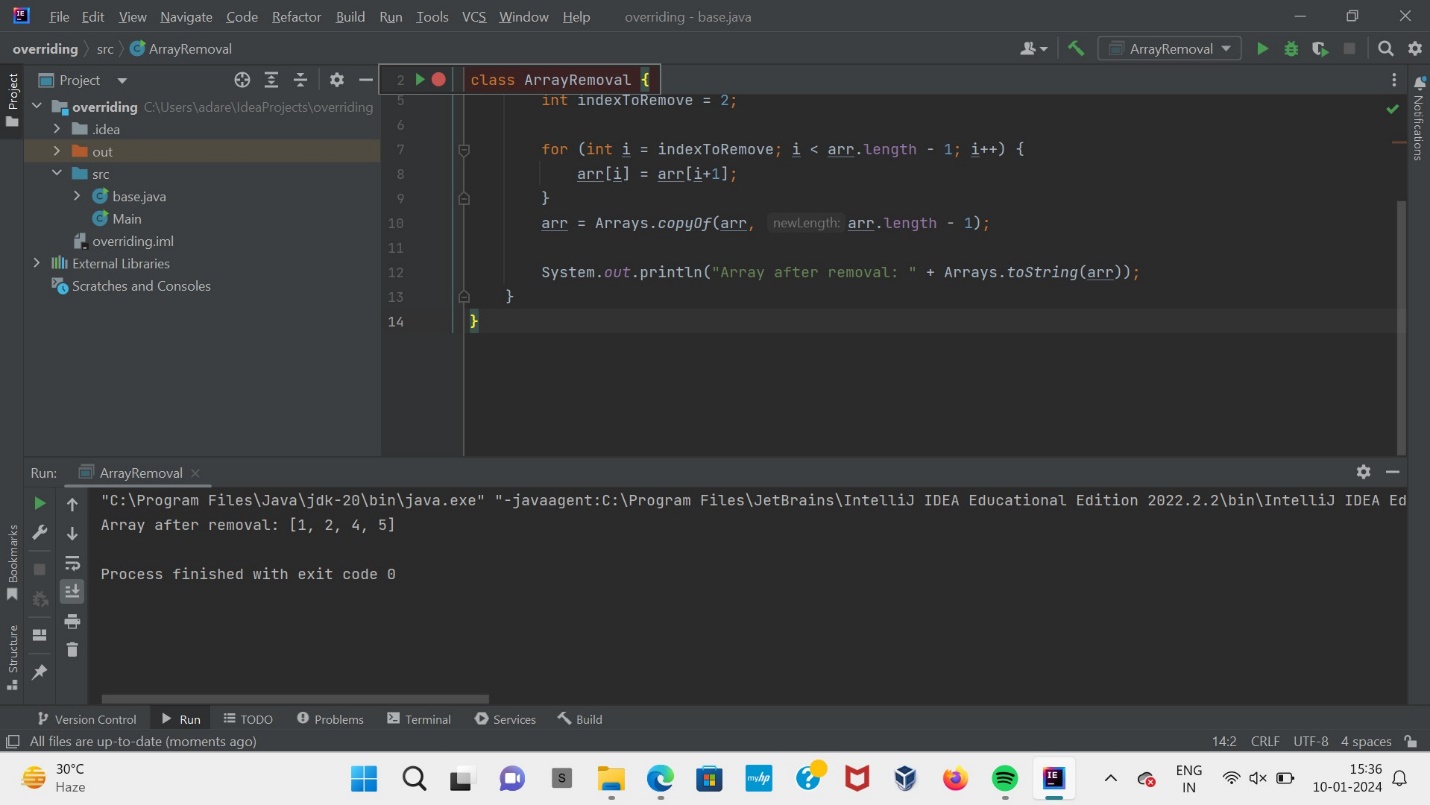
arr = Arrays.copyOf(arr, arr.length - 1);

System.out.println("Array after removal: " + Arrays.toString(arr));

}

}

Output:



Program:5

import java.util.Arrays;

class ArrayRemoval {

public static void main(String[] args) {

int[] arr = {1, 2, 3, 4, 5};

int indexToRemove = 2;

for (int i = indexToRemove; i < arr.length - 1; i++) {

arr[i] = arr[i+1];

}

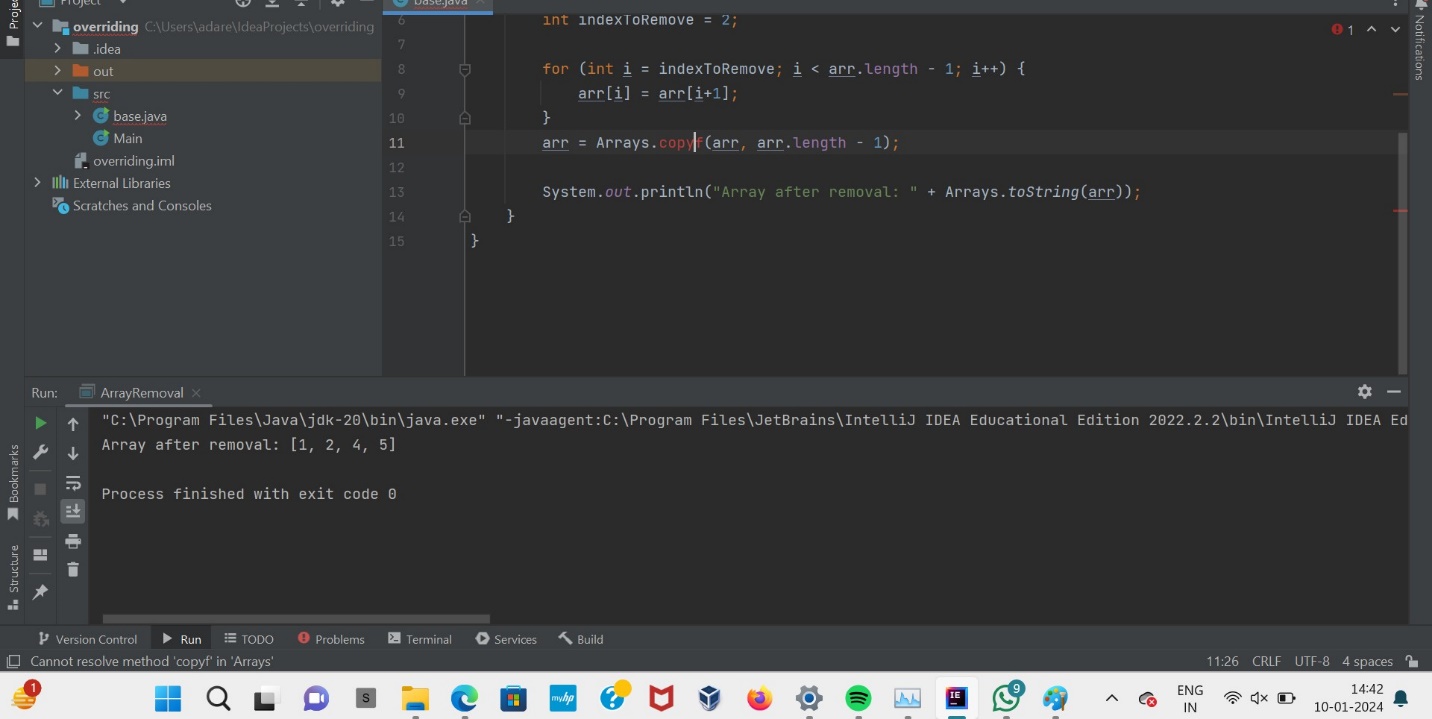
arr = Arrays.copyOf(arr, arr.length - 1);

System.out.println("Array after removal: " + Arrays.toString(arr));

}

}

Output:



Program:6

import java.util.ArrayList;

class ArrayListCheckEmpty {

public static void main(String[] args) {

ArrayList<Integer> list1 = new ArrayList<Integer>();

ArrayList<Integer> list2 = new ArrayList<Integer>();

list1.add(1);

list1.add(2);

list1.add(3);

if (list1.isEmpty()) {

System.out.println("list1 is empty");

} else {

System.out.println("list1 is not empty");

}

if (list2.isEmpty()) {

System.out.println("list2 is empty");

} else {

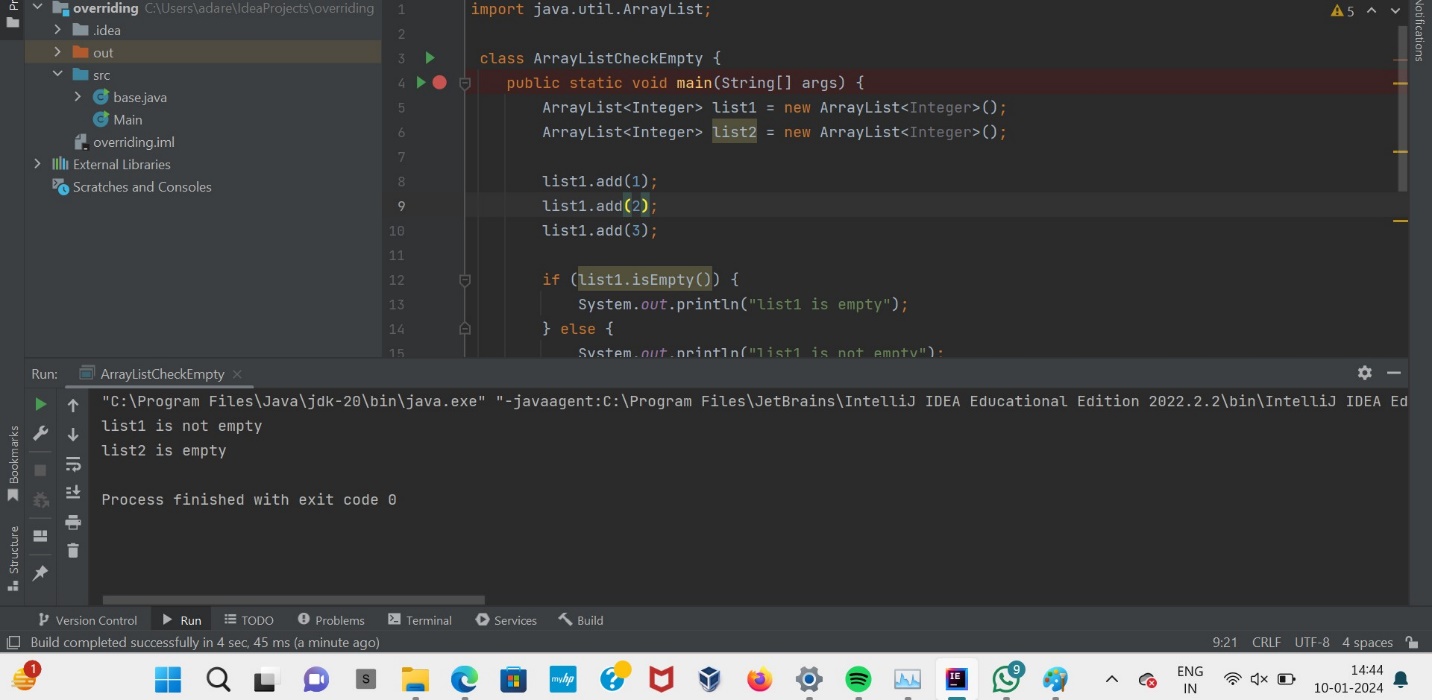
System.out.println("list2 is not empty");

}

}

}

Output:



Program:7

import java.io.\*;

class Number {

Number()

{

super();

System.out.println("Constructor Called");

}

public static void main(String[] args)

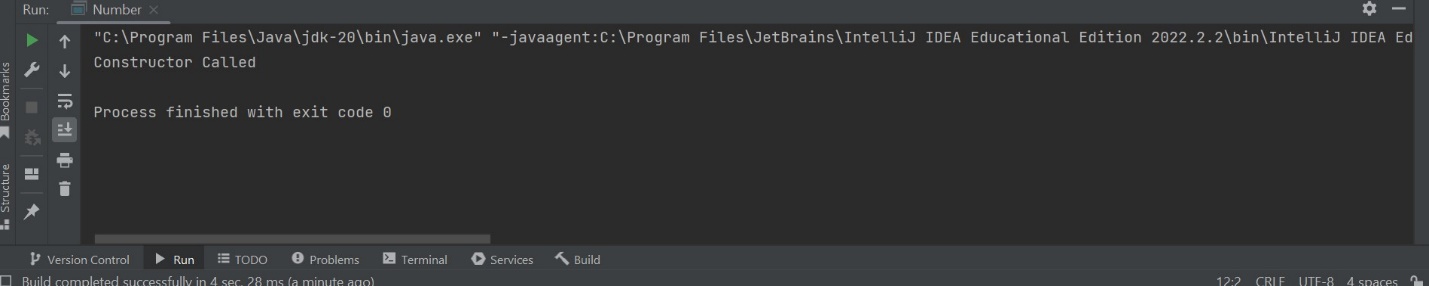
{

Number num = new Number();

}

}

Output:



Program:8

import java.util.LinkedList;

import java.util.Queue;

class QueueExample {

public static void main(String[] args) {

Queue<String> queue = new LinkedList<String>();

queue.add("A");

queue.add("B");

queue.add("C");

queue.add("D");

queue.add("E");

while (!queue.isEmpty()) {

System.out.println(queue.remove());

}

}

}

Output:

