EC4070: Data Structures and Algorithms

LAB 03

FINAL

K.J.M.U.G.S. Eranda Jayasinghe

2021/E/075

SEMESTER 4

EC4070

11.10.2023

Q1.a.

```
import java.util.Stack;


public class Stack1 {
   public static void main(String[] args) {
      Stack<String> stack = new Stack<>();


      stack.push("cat");
      stack.push("dog");
      stack.push("ant");


      System.out.print("Stack elements: ");
      System.out.println(stack);


      String topn = stack.peek();
      System.out.println("Peek (top): " + topn);


      String popn = stack.pop();
      System.out.println("Pop: " + popn);


      System.out.print("Stack after pop: ");
      System.out.println(stack);


      int sizen = stack.size();
      System.out.println("Stack size: " + sizen);
   }
}
```

Stack1.java | Word1.java

```java
import java.util.Stack;

public class Stack1 {
    public static void main(String[] args) {
        Stack<String> stack = new Stack<>();

        stack.push("cat");
        stack.push("dog");
        stack.push("ant");

        System.out.print("Stack elements: ");
        System.out.println(stack);

        String topn = stack.peek();
        System.out.println("Peek (top): " + topn);

        String popn = stack.pop();
        System.out.println("Pop: " + popn);

        System.out.print("Stack after pop: ");
        System.out.println(stack);

        int sizen = stack.size();
        System.out.println("Stack size: " + sizen);
    }
}
```

```
C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>java Stack1
Stack elements: [cat, dog, ant]
Peek (top): ant
Pop: ant
Stack after pop: [cat, dog]
Stack size: 2

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>
```

Q1. b.

```
import java.util.Scanner;
import java.util.Stack;

public class Word1{
	public static void main(String[] args){
		Scanner x = new Scanner(System.in);
		System.out.print("Enter your word : ");
		String word = x.nextLine();

		Stack<Character> stack = new Stack<>();

		for(int i=0; i < word.length();i++){
			stack.push(word.charAt(i));
		}
		System.out.print("Your reverced word is ");
		while(!stack.isEmpty()){
			System.out.print(stack.pop());
		}
	}
}
```

Stack1.java | Word1.java

```java
import java.util.Scanner;
import java.util.Stack;

public class Word1{
    public static void main(String[] args){
        Scanner x = new Scanner(System.in);
        System.out.print("Enter your word : ");
        String word = x.nextLine();

        Stack<Character> stack = new Stack<>();

        for(int i=0; i < word.length();i++){
            stack.push(word.charAt(i));
        }
        System.out.print("Your reverced word is ");
        while(!stack.isEmpty()){
            System.out.print(stack.pop());
        }
    }
}
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.22000.2416]
(c) Microsoft Corporation. All rights reserved.

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>javac Word1.java

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>java Word1
Enter your word : EranDa
Your reverced word is aDnarE
C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>
```

Q1.b2.

```
import java.util.Scanner;
import java.util.Stack;

public class Delim {
   public static void main(String[] args) {
      Scanner x = new Scanner(System.in);
      System.out.print("Enter delimiter word: ");
      String str = x.nextLine();

      boolean isMatched = checkDM(str);

      if (isMatched) {
         System.out.println("Delimiters matched.");
      } else {
         System.out.println("Delimiters are not matched.");
      }
   }

   public static boolean checkDM(String str) {
      Stack<Character> stack = new Stack<>();

      for (int i = 0; i < str.length(); i++) {
         char c = str.charAt(i);

         if (c == '(' || c == '[' || c == '{') {
            stack.push(c);
         }
                        else if (c == ')' || c == ']' || c == '}') {
            if (stack.isEmpty()) {
```

```
                return false;
            }

            char open = stack.pop();

            if (!isMatchingPair(open, c)) {
                return false;
            }
        }
    }

    return stack.isEmpty();
  }

  public static boolean isMatchingPair(char open, char close) {
    return (open == '(' && close == ')') || (open == '[' && close == ']') || (open == '{' && close == '}');
  }
}
```

Stack1.java | Word1.java | Delim.java

```java
import java.util.Scanner;
import java.util.Stack;

public class Delim {
    public static void main(String[] args) {
        Scanner x = new Scanner(System.in);
        System.out.print("Enter delimiter word: ");
        String str = x.nextLine();

        boolean isMatched = checkDM(str);

        if (isMatched) {
            System.out.println("Delimiters matched.");
        } else {
            System.out.println("Delimiters are not matched.");
        }
    }

    public static boolean checkDM(String str) {
        Stack<Character> stack = new Stack<>();

        for (int i = 0; i < str.length(); i++) {
            char c = str.charAt(i);

            if (c == '(' || c == '[' || c == '{') {
                stack.push(c);
            }
            else if (c == ')' || c == ']' || c == '}') {
                if (stack.isEmpty()) {
                    return false;
                }

                char open = stack.pop();

                if (!isMatchingPair(open, c)) {
                    return false;
                }
            }
        }

        return stack.isEmpty();
    }

    public static boolean isMatchingPair(char open, char close) {
        return (open == '(' && close == ')') || (open == '[' && close == ']') || (open == '{' && close == '}');
    }
}

```

Select C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.22000.2416]
(c) Microsoft Corporation. All rights reserved.

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>javac Delim.java

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>java Delim
Enter delimiter word: {er(n[da])}
Delimiters matched.

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>javac Delim.java

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>java Delim
Enter delimiter word: {er)rtrt)
Delimiters are not matched.

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>
```

Q2.a.

```
import java.util.LinkedList;
import java.util.Queue;

public class Queue1 {
    public static void main(String[] args) {
        Queue<Integer> queue = new LinkedList<>();

        for (int i = 1; i <= 5; i++) {
            queue.offer(i);
        }

        displayQueue(queue);

        int removedItem = queue.poll();
        System.out.println("Dequeued item: " + removedItem);

        displayQueue(queue);

        int frontItem = queue.peek();
        System.out.println("Front item: " + frontItem);

        boolean isEmpty = queue.isEmpty();
        System.out.println("Is the queue empty? " + isEmpty);
    }

    public static void displayQueue(Queue<Integer> queue) {
        System.out.print("Queue: ");
        for (int item : queue) {
            System.out.print(item + " ");
```

```
    }

    System.out.println();

  }

}
```

Stack1.java | Word1.java | Delim.java | Queue1.java

```java
import java.util.LinkedList;
import java.util.Queue;

public class Queue1{
    public static void main(String[] args) {
        Queue<Integer> queue = new LinkedList<>();

        for (int i = 1; i <= 5; i++) {
            queue.offer(i);
        }

        displayQueue(queue);

        int removedItem = queue.poll();
        System.out.println("Dequeued item: " + removedItem);

        displayQueue(queue);

        int frontItem = queue.peek();
        System.out.println("Front item: " + frontItem);

        boolean isEmpty = queue.isEmpty();
        System.out.println("Is the queue empty? " + isEmpty);
    }

    public static void displayQueue(Queue<Integer> queue) {
        System.out.print("Queue: ");
        for (int item : queue) {
            System.out.print(item + " ");
        }
        System.out.println();
    }
}
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.22000.2416]
(c) Microsoft Corporation. All rights reserved.

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>javac Word1.java

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>java Word1
Enter your word : EranDa
Your reverced word is aDnarE
C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>javac Stack1.java

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>java Stack1
Stack elements: [cat, dog, ant]
Peek (top): ant
Pop: ant
Stack after pop: [cat, dog]
Stack size: 2

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>javac Queue1.java

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>java Queue1
Queue: 1 2 3 4 5
Dequeued item: 1
Queue: 2 3 4 5
Front item: 2
Is the queue empty? false

C:\Users\2021E075\OneDrive - University of Jaffna\lab3 com>
```

Q2.b

```
import java.util.Queue;
import java.util.PriorityQueue;
import java.util.Collections;

public class priority{
	public static void main(String[] args){
		Queue<String> queue=new PriorityQueue<>();

		queue.offer("dog");
		queue.offer("cat");
		queue.offer("elephant");
		queue.offer("bird");
		queue.offer("cow");
		System.out.println(queue);
		Queue<Integer> queue2 = new PriorityQueue<>(Collections.reverseOrder());

		queue2.offer(2);
		queue2.offer(9);
		queue2.offer(101);
		queue2.offer(89);
		queue2.offer(5);
		System.out.println(queue2);
		queue.poll();
		queue.poll();
		System.out.println(queue.poll());
		System.out.println(queue.peek());
		System.out.println(queue);
		System.out.println(queue.isEmpty());
		System.out.println(queue.size());
```

```
    }

    }
```

C:\Users\erand\OneDrive - University of Jaffna\lab3 com\priority.java - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Stack1.java | Word1.java | Delim.java | Queue1.java | priority.java

```java
import java.util.Queue;
import java.util.PriorityQueue;
import java.util.Collections;

public class priority{
    public static void main(String[] args){
        Queue<String> queue=new PriorityQueue<>();

        queue.offer("dog");
        queue.offer("cat");
        queue.offer("elephant");
        queue.offer("bird");
        queue.offer("cow");
        System.out.println(queue);
        Queue<Integer> queue2 = new PriorityQueue<>(Collections.reverseOrder());

        queue2.offer(2);
        queue2.offer(9);
        queue2.offer(101);
        queue2.offer(89);
        queue2.offer(5);
        System.out.println(queue2);
        queue.poll();
        queue.poll();
        System.out.println(queue.poll());
        System.out.println(queue.peek());
        System.out.println(queue);
        System.out.println(queue.isEmpty());
        System.out.println(queue.size());
        }
        }
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.19045.3516]
(c) Microsoft Corporation. All rights reserved.

C:\Users\erand\OneDrive - University of Jaffna\lab3 com>javac priority.java

C:\Users\erand\OneDrive - University of Jaffna\lab3 com>java priority
[bird, cat, elephant, dog, cow]
[101, 89, 9, 2, 5]
cow
dog
[dog, elephant]
false
2

C:\Users\erand\OneDrive - University of Jaffna\lab3 com>
```

Q3.

```
import java.util.LinkedList;


public class List1 {
   public static void main(String[] args) {
      LinkedList<String> list = new LinkedList<>();


      list.push("dog");
      list.push("cat");
      list.push("elephant");
      list.offer("cow");
      list.push("bird");


      list.poll();
      list.pop();


      System.out.println("Current List: " + list);


      list.add(3, "monkey");


      list.remove("dog");


      System.out.println("Index of 'dog': " + list.indexOf("dog"));


      System.out.println("Current List: " + list);


      System.out.println("Peek First: " + list.peekFirst());
      System.out.println("Peek Last: " + list.peekLast());


      list.addFirst("car");
```

```
        list.addLast("van");

        System.out.println("Updated List: " + list);

        String first = list.removeFirst();
        String last = list.removeLast();

        System.out.println("Final List: " + list);

        System.out.println("Removed First: " + first);
        System.out.println("Removed Last: " + last);
    }
}
```

Stack1.java | Word1.java | Delim.java | Queue1.java | priority.java | List1.java

```java
import java.util.LinkedList;

public class List1 {
    public static void main(String[] args) {
        LinkedList<String> list = new LinkedList<>();

        list.push("dog");
        list.push("cat");
        list.push("elephant");
        list.offer("cow");
        list.push("bird");

        list.poll();
        list.pop();

        System.out.println("Current List: " + list);

        list.add(3, "monkey");

        list.remove("dog");

        System.out.println("Index of 'dog': " + list.indexOf("dog"));

        System.out.println("Current List: " + list);

        System.out.println("Peek First: " + list.peekFirst());
        System.out.println("Peek Last: " + list.peekLast());

        list.addFirst("car");
        list.addLast("van");

        System.out.println("Updated List: " + list);

        String first = list.removeFirst();
        String last = list.removeLast();

        System.out.println("Final List: " + list);

        System.out.println("Removed First: " + first);
        System.out.println("Removed Last: " + last);
    }
}
```

```
C:\Users\erand\OneDrive - University of Jaffna\lab3 com>javac List1.java

C:\Users\erand\OneDrive - University of Jaffna\lab3 com>java List1
Current List: [cat, dog, cow]
Index of 'dog': -1
Current List: [cat, cow, monkey]
Peek First: cat
Peek Last: monkey
Updated List: [car, cat, cow, monkey, van]
Final List: [cat, cow, monkey]
Removed First: car
Removed Last: van

C:\Users\erand\OneDrive - University of Jaffna\lab3 com>
```