Assignment 5

21AIE111

Data Structure and Algorithms – SEM-II

Professor – Dr. Sachin Sir

Submitted By: Vikhyat Bansal [CB.EN.U4AIE21076]

AMRITA VISHWA VIDYAPEETHAM

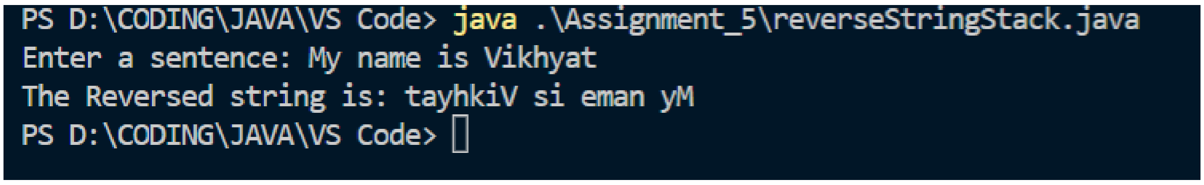
DEEMED TO BE UNIVERSITY

1. Write a java code to reverse a string using stack.

CODE:

```java
package Assignment_5;
import java.util.*;
import java.io.*;
import java.io.IOException;

    /**first element of string must go first in stack so that the
last element of string is at top of the stack so when elements are
popped from the stack the values in the string in such a manner that
it reverses the input string.*/

public class reverseStringStack {
   private String str;
   private String output;
   public reverseStringStack(String in) {
      str = in;
   }
   public String doReverse() {
      int stackSize = str.length();
      Stack theStack = new Stack(stackSize);

      for (int i = 0; i < str.length(); i++) {
         char ch = str.charAt(i);
         theStack.push(ch);
      }
      output = "";
      while (!theStack.isEmpty()) {
         char ch = theStack.pop();
         output = output + ch;
      }
      return output;
   }
   public static void main(String[] args) throws IOException {
      System.out.print("Enter a sentence: ");
      Scanner scan = new Scanner(System.in);
      String str = scan.nextLine();
      String output;
      reverseStringStack theReverser =
      new reverseStringStack(str);
      output = theReverser.doReverse();
      System.out.println("The Reversed string is: " + output);
   }
   class Stack {
      private int maxSize;
      private char[] stackArray;
      private int top;

      public Stack(int max) {
         maxSize = max;
         stackArray = new char[maxSize];
         top = -1;
      }
      public void push(char j) {
         stackArray[++top] = j;
      }
      public char pop() {
         return stackArray[top--];
      }
      public char peek() {
         return stackArray[top];
      }
      public boolean isEmpty() {
         return (top == -1);
      }
   }
}
```

OUTPUT:



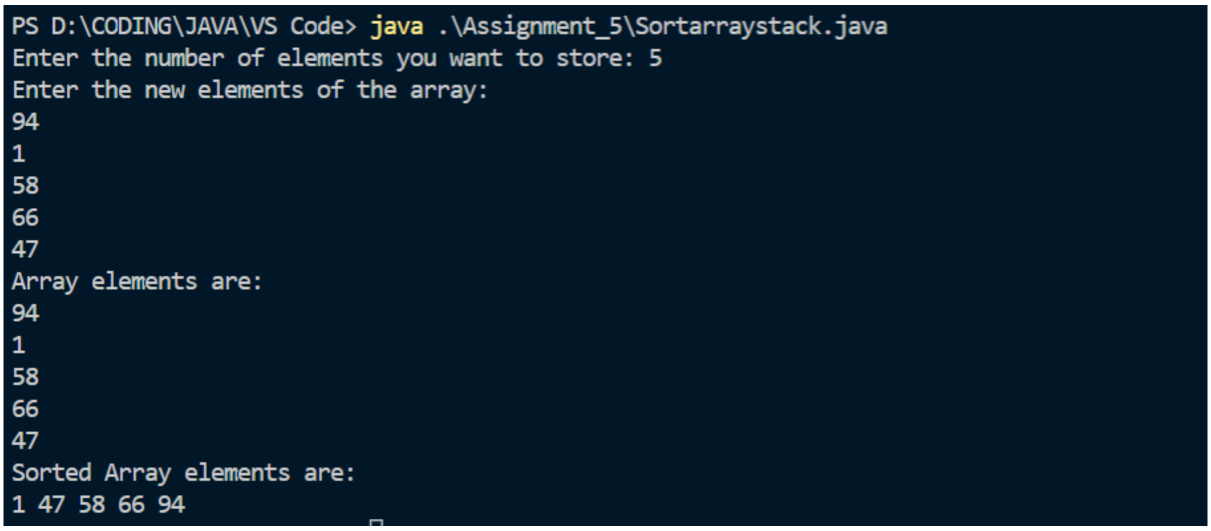
```
PS D:\CODING\JAVA\VS Code> java .\Assignment_5\reverseStringStack.java
Enter a sentence: My name is Vikhyat
The Reversed string is: tayhkiV si eman yM
PS D:\CODING\JAVA\VS Code> 
```

2. Write a java code to sort an array using stack.

CODE:

```java
package Assignment_5;

import java.io.*;
import java.util.*;
import java.util.Scanner;
public class Sortarraystack {



      static Stack<Integer> sortStack(Stack<Integer> input){
          Stack<Integer> tmpStack = new Stack<Integer>();

          while(!input.empty()){
              int tmp = input.peek();
              input.pop();

              while(!tmpStack.empty() && tmpStack.peek() < tmp){
                  input.push(tmpStack.peek());
                  tmpStack.pop();
              }

              tmpStack.push(tmp);
          }
          return tmpStack;
      }

      static void sortUsingStack(int []arr, int n){

          Stack<Integer> input = new Stack<Integer>();

          for(int i = 0; i < n; i++){
              input.push(arr[i]);
          }

          Stack<Integer> tmpStack = sortStack(input);

          for(int i = 0; i < n; i++){
              arr[i] = tmpStack.peek();
              tmpStack.pop();
          }
      }



    public static void main(String[] args){

        int n;
        try (Scanner sc = new Scanner(System.in)) {
            System.out.print("Enter the number of elements you want
to store: ");
            //reading the number of elements from the that we want
to enter
            n=sc.nextInt();
            int arr[] = new int[n];

            System.out.println("Enter the new elements of the array:
");
            for(int i=0; i<n; i++)
            {
            //reading array elements from the user
            arr[i]=sc.nextInt();
            }
            System.out.println("Array elements are: ");
            for (int j=0; j<n; j++)
      {
         System.out.println(arr[j]);
      }

      sortUsingStack(arr, n);

      System.out.println("Sorted Array elements are: ");
        for(int i = 0; i < n; i++){
            System.out.print(arr[i] + " ");
        }
        }

    class Stack {
        private int maxSize;
        private char[] stackArray;
        private int top;
        public Stack(int max) {
           maxSize = max;
           stackArray = new char[maxSize];
           top = -1;
        }
        public void push(char j) {
           stackArray[++top] = j;
        }
        public char pop() {
           return stackArray[top--];
        }
        public char peek() {
           return stackArray[top];
        }
        public boolean isEmpty() {
           return (top == -1);
        }


}
}
}
```

OUTPUT:



```
PS D:\CODING\JAVA\VS Code> java .\Assignment_5\Sortarraystack.java
Enter the number of elements you want to store: 5
Enter the new elements of the array:
94
1
58
66
47
Array elements are:
94
1
58
66
47
Sorted Array elements are:
1 47 58 66 94
```

3. Implement stack using queue.

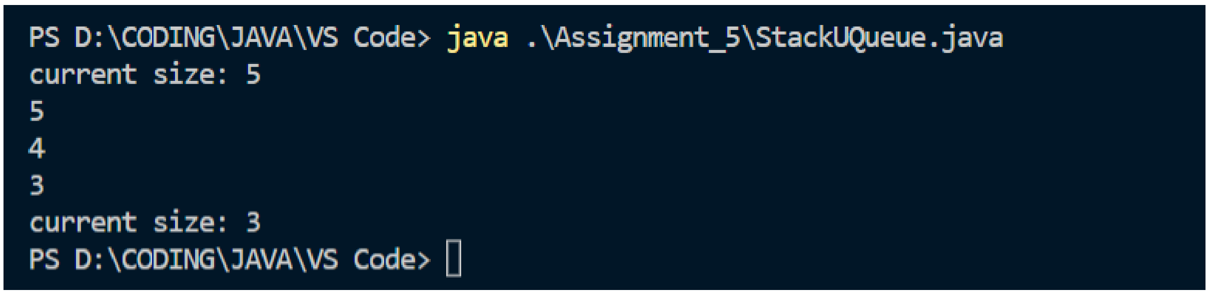
CODE:

```
package Assignment_5;

/* Java Program to implement a stack using
two queue */
import java.util.*;

class StackUQueue {

    public static void main(String[] args)
    {
        Stack s = new Stack();
        s.push(1);
        s.push(2);
        s.push(3);
        s.push(4);
        s.push(5);

        System.out.println("current size: " + s.size());
        System.out.println(s.top());
        s.pop();
        System.out.println(s.top());
        s.pop();
        System.out.println(s.top());

        System.out.println("current size: " + s.size());
    }

    static class Stack {
        // Two inbuilt queues q1 and q2 are used to implement stack
        static Queue<Integer> q1 = new LinkedList<Integer>();
        static Queue<Integer> q2 = new LinkedList<Integer>();

        // To maintain current number of elements in stack

        static int curr_size;

        Stack()
        {
```

```
        curr_size = 0;
    }

    static void push(int x)
    {
        curr_size++;

        // Push x first in empty q2
        q2.add(x);

        // Push all the remaining
        // elements in q1 to q2.
        while (!q1.isEmpty()) {
            q2.add(q1.peek());
            q1.remove();
        }

        // swap the names of two queues
        Queue<Integer> q = q1;
        q1 = q2;
        q2 = q;
    }

    static void pop()
    {

        // if no elements are there in q1
        if (q1.isEmpty())
            return;
        q1.remove();
        curr_size--;
    }

    static int top()
    {
        if (q1.isEmpty())
            return -1;
        return q1.peek();
    }

    static int size()
    {
```

```java
                return curr_size;
            }
        }


}
```

OUTPUT:



```
PS D:\CODING\JAVA\VS Code> java .\Assignment_5\StackUQueue.java
current size: 5
5
4
3
current size: 3
PS D:\CODING\JAVA\VS Code> 
```

4. Is it possible to use an array to implement two stack. If yes, then show the implementation.

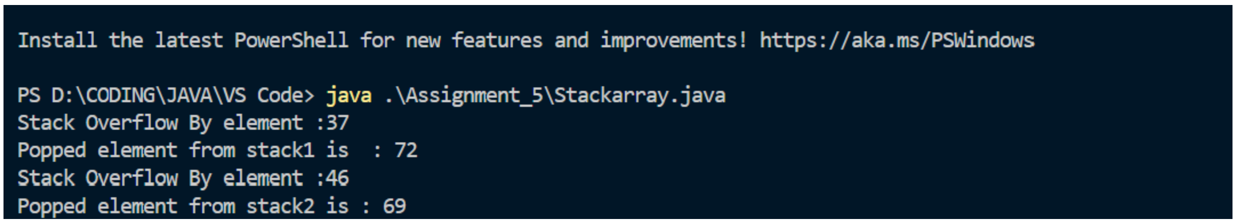
CODE:

```
import java.util.*;

public class Stackarray {
    public static void main(String[] args)
{
    twoStacks ts = new twoStacks(5);
    ts.push1(55);
    ts.push2(9);
    ts.push2(69);
    ts.push1(72);
    ts.push2(37);
    System.out.print("Popped element from stack1 is "
                    + " : " + ts.pop1() +"\n");
    ts.push2(46);
    System.out.print("Popped element from stack2 is "
                    + ": " + ts.pop2()
                    +"\n");
}
}
class twoStacks
{

int[] arr;
int size;
int top1, top2;

// Constructor
twoStacks(int n)
{
    size = n;
    arr = new int[n];
    top1 = n / 2 + 1;
    top2 = n / 2;
}

// Method to push an element x to stack1
void push1(int x)
```

```java
{

    // There is at least one empty
    // space for new element
    if (top1 > 0)
    {
    top1--;
    arr[top1] = x;
    }
    else
    {
    System.out.print("Stack Overflow"
                    + " By element :" + x +"\n");
    return;
    }
}

// Method to push an element
// x to stack2
void push2(int x)
{

    // There is at least one empty
    // space for new element
    if (top2 < size - 1)
    {
    top2++;
    arr[top2] = x;
    }
    else
    {
    System.out.print("Stack Overflow"
                    + " By element :" + x +"\n");
    return;
    }
}

// Method to pop an element from first stack
int pop1()
{
    if (top1 <= size / 2)
    {
```

```java
    int x = arr[top1];
    top1++;
    return x;
    }
    else
    {
    System.out.print("Stack UnderFlow");
    System.exit(1);
    }
    return 0;
}

// Method to pop an element
// from second stack
int pop2()
{
    if (top2 >= size / 2 + 1)
    {
    int x = arr[top2];
    top2--;
    return x;
    }
    else
    {
    System.out.print("Stack UnderFlow");
    System.exit(1);
    }
    return 1;
}
}
```

OUTPUT:



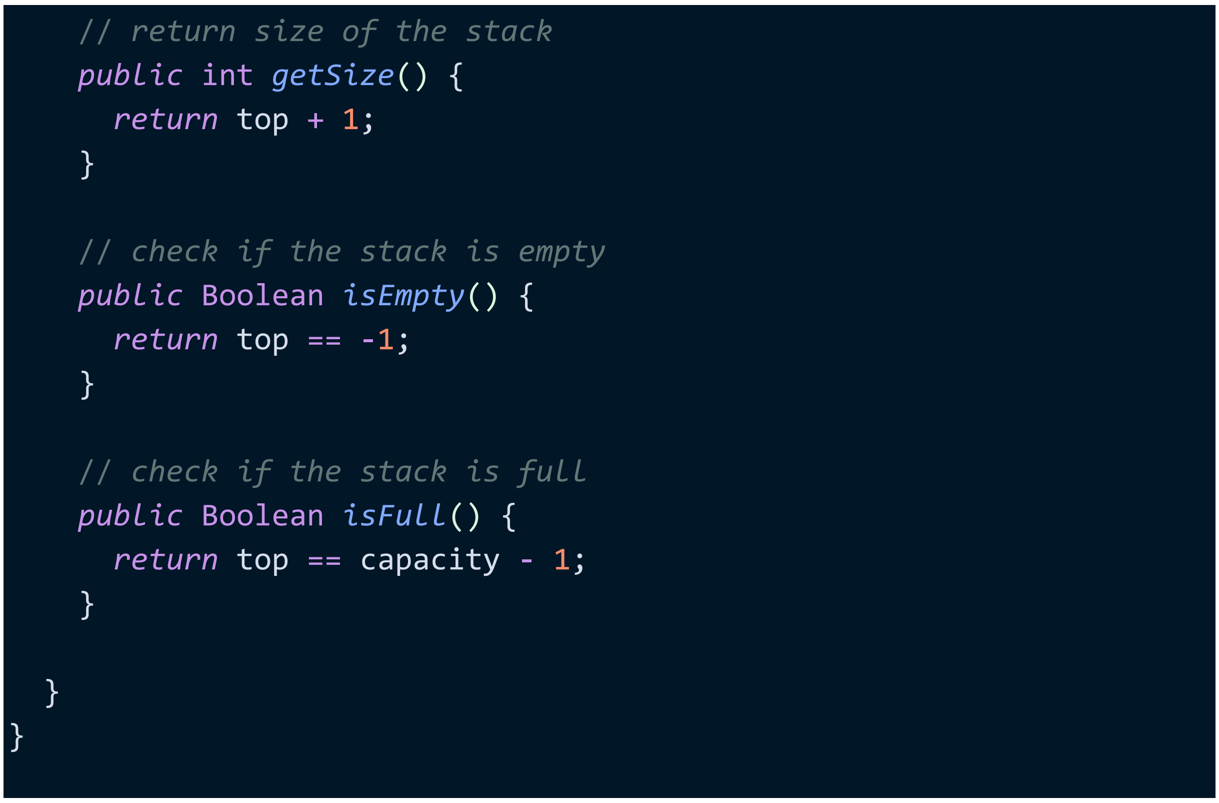
```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\CODING\JAVA\VS Code> java .\Assignment_5\Stackarray.java
Stack Overflow By element :37
Popped element from stack1 is  : 72
Stack Overflow By element :46
Popped element from stack2 is : 69
```

5. Implement queue using stack.

CODE:

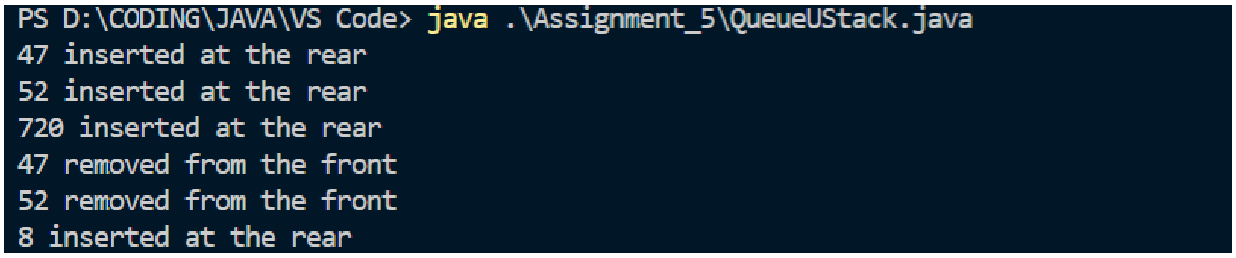
```
import java.util.*;
import java.io.*;

// Stack implementation in Java
public class QueueUStack {

Stack stack1 = new Stack(10);
Stack stack2 = new Stack(10);

   public void enqueue(int element) {
      stack1.push(element);
      System.out.println(element + " inserted at the rear
location");
   }

   public void dequeue() {
      if(stack2.isEmpty()) {
         while (!stack1.isEmpty()) {
            stack2.push(stack1.pop());
         }
      }
      System.out.println(stack2.pop() + " removed from the front
location");
   }
   public static void main(String args[]) {
      QueueUStack Q = new QueueUStack();
      Q.enqueue(47);
      Q.enqueue(52);
      Q.enqueue(720);
      Q.dequeue();
      Q.dequeue();
      Q.enqueue(8);
   }

   class Stack {

    // store elements of stack
    private int arr[];
```

```java
  // represent top of stack
  private int top;
  // total capacity of the stack
  private int capacity;

  // Creating a stack
  Stack(int size) {
    // initialize the array
    // initialize the stack variables
    arr = new int[size];
    capacity = size;
    top = -1;
  }

  // push elements to the top of stack
  public void push(int x) {
    if (isFull()) {
      System.out.println("Stack OverFlow");

      // terminates the program
      System.exit(1);
    }

    // insert element on top of stack
    arr[++top] = x;
  }

  // pop elements from top of stack
  public int pop() {

    // if stack is empty
    // no element to pop
    if (isEmpty()) {
      System.out.println("STACK EMPTY");
      // terminates the program
      System.exit(1);
    }

    // pop element from top of stack
    return arr[top--];
  }
```



```java
    // return size of the stack
    public int getSize() {
      return top + 1;
    }

    // check if the stack is empty
    public Boolean isEmpty() {
      return top == -1;
    }

    // check if the stack is full
    public Boolean isFull() {
      return top == capacity - 1;
    }

  }
}
```

OUTPUT:



```
PS D:\CODING\JAVA\VS Code> java .\Assignment_5\QueueUStack.java
47 inserted at the rear
52 inserted at the rear
720 inserted at the rear
47 removed from the front
52 removed from the front
8 inserted at the rear
```

THANK YOU