

DATA structure lab 03



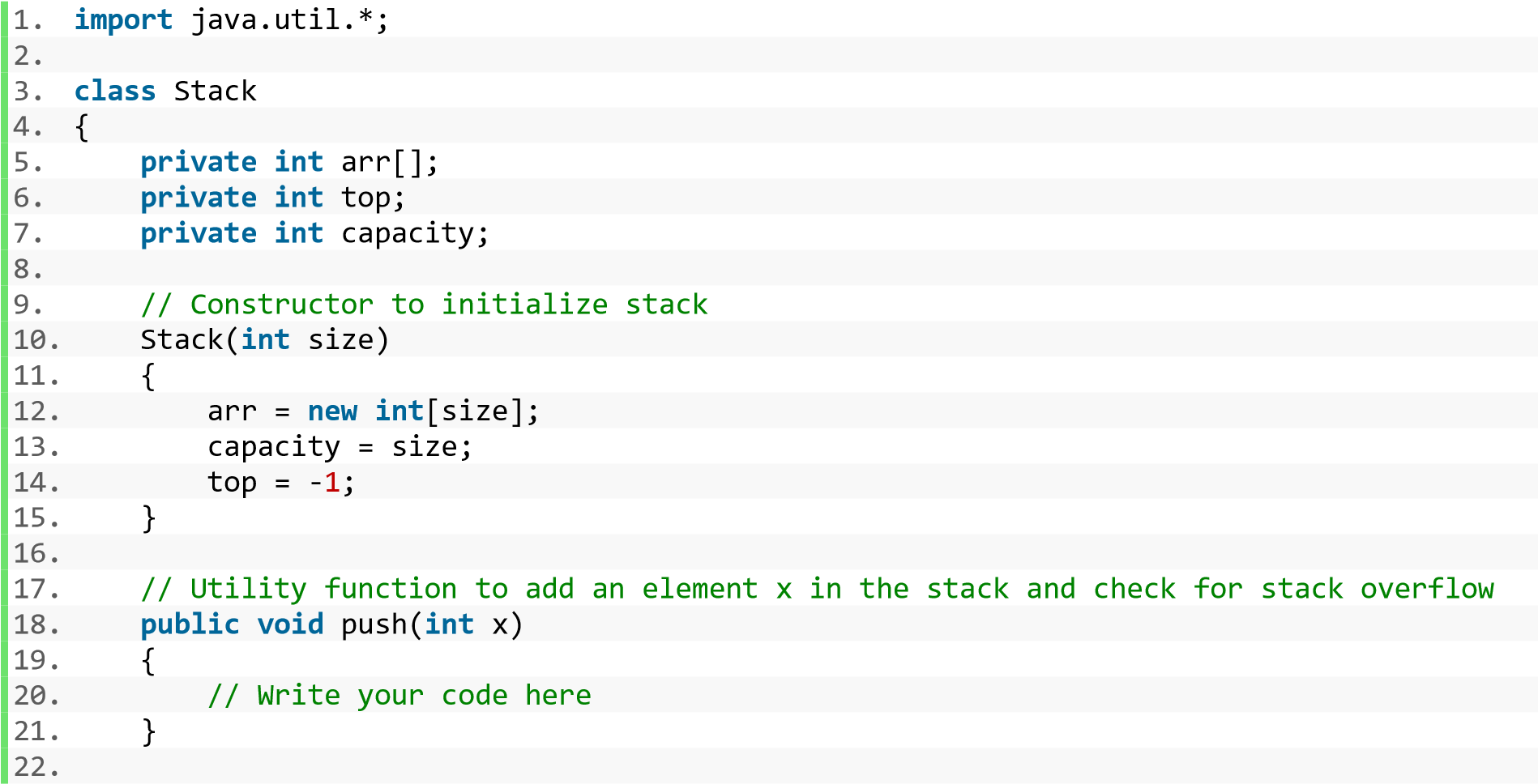
Lab 03- Stacks,

# Stack

Note: Keep this code with you till the course ends.

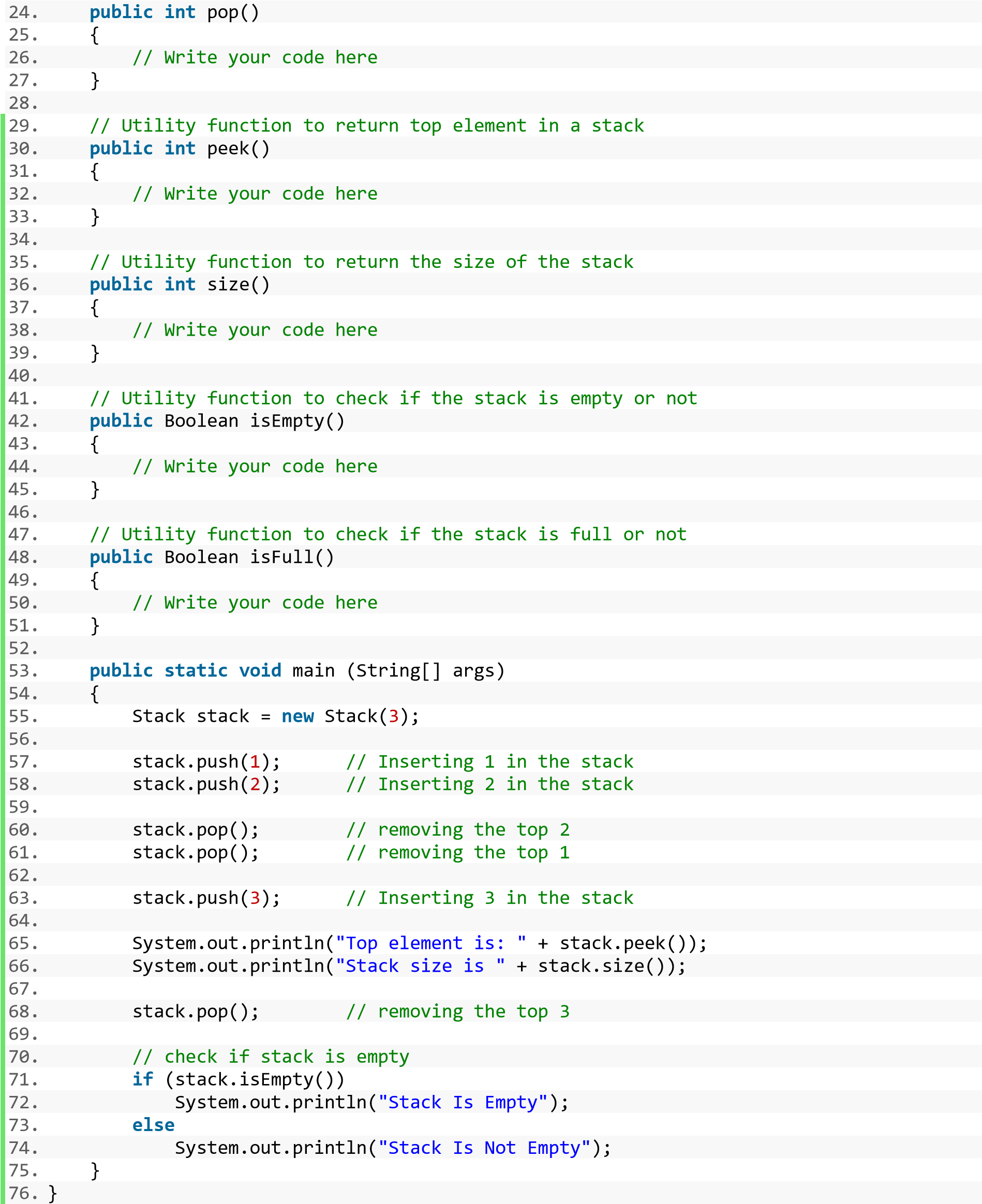
**Task 01: (Stack using array)**

**Understand provided code and implement all required methods in Stack. Stack Code is given below:**

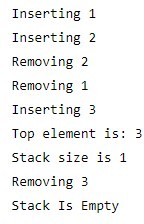


23. // Utility function to pop top element from the stack and check for stack underflow

Data Structures & Applications by Saif Hassan Sukkur IBA University



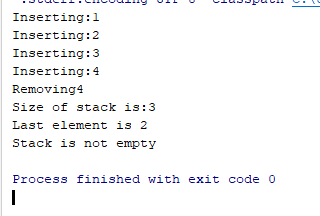
After implementing all the methods, run the code. Your output should be like as follows:



CODE IMPLEMENTATION

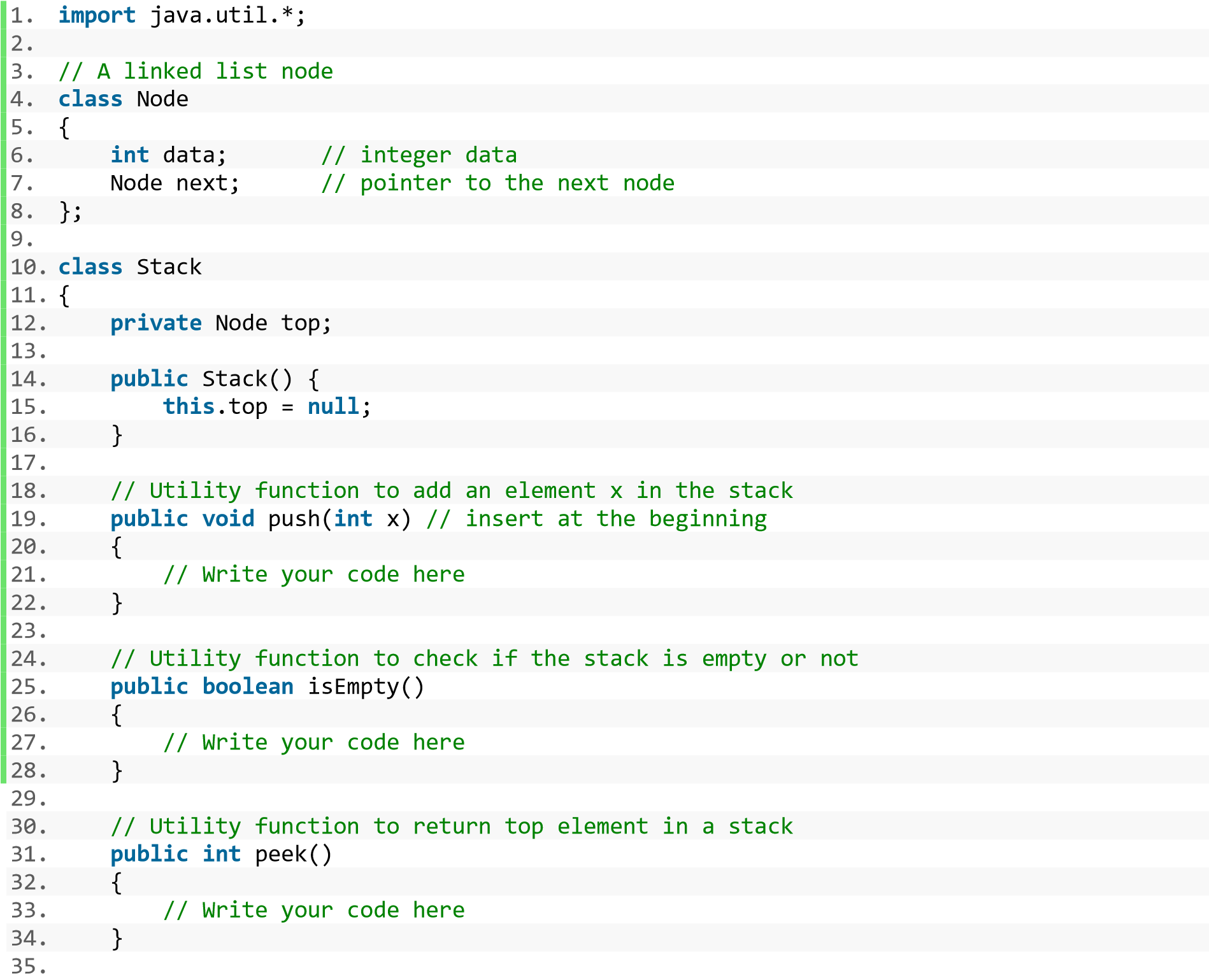
package StackLabTasks;  
  
public class Stack {  
  
 private int arr[];  
 private int top;  
 private int capacity;  
  
  
 *//Constructor to initialize stack* Stack(int size){  
  
 arr=new int[size];  
 capacity=size;  
 top=-1;  
  
  
 }  
  
 *//add data in the stack and check for overflow of stack* public void push(int data){  
  
 System.*out*.print("Inserting:");  
 System.*out*.println(arr[++top]=data);  
  
  
 }  
  
*//pop top elements from mistake and check for overflow of stack  
 //pop remove the top element in stack* public void pop(){  
  
 System.*out*.println("Removing"+arr[top--]);  
  
 }  
  
 *//peek return last element* public void peek(){  
  
 System.*out*.println("Last element is "+top);  
  
  
 }  
  
 public int size(){  
  
 return top+1;  
  
 }  
  
  
  
  
  
  
  
 public Boolean isEmpty(){  
  
 return top==-1;  
  
 }  
  
  
 public Boolean isFull(){  
  
  
 return top==capacity-1;  
  
 }  
  
  
 public static void main(String[] args) {  
 Stack stack1=new Stack(4);  
 stack1.push(1);  
 stack1.push(2);  
 stack1.push(3);  
 stack1.push(4);  
 stack1.pop();  
  
  
 System.*out*.print("Size of stack is:");  
 System.*out*.println(stack1.size());  
  
 stack1.peek();  
  
 if(stack1.isEmpty()){  
  
 System.*out*.println("Stack is empty");  
 }  
 else {  
 System.*out*.println("Stack is not empty");  
 }  
  
  
  
  
  
  
  
  
  
  
  
  
 }  
}

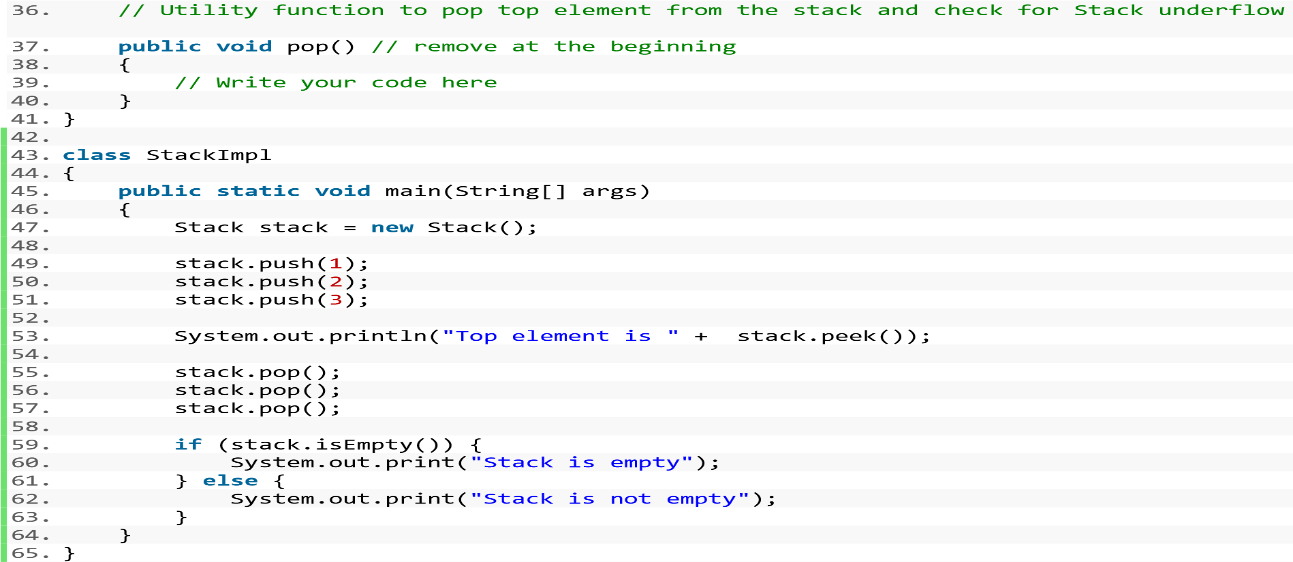
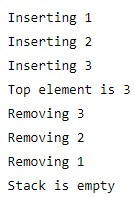
OUTPUT:



**Task 02:(Stack using Linked list)**

**Understand provided code and implement all required methods in Stack. Stack Code is given below:**

****

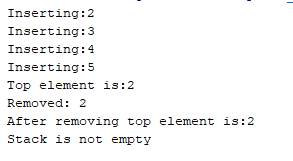
****

After implementing all the methods, run the code. Your output should be like as follows:

CODE IMPLEMENTATION

package StackLabTasks;  
  
public class StackInLinkedList {  
  
 private Node top;  
 private int length;  
  
 public class Node{  
  
 private int data;  
 private Node next;  
  
  
 public Node(int data){  
  
 this.data=data;  
  
  
 }  
  
 }  
 *//class constructor* public StackInLinkedList(){  
  
 top=null;  
 length=0;  
  
  
 }  
  
 public void push(int data){  
 Node newNode=new Node(2); *//new node created with data 2* newNode.next=top; *// pointer is null till now* top=newNode; *//new node and top is pointing to same data* length++;  
 System.*out*.println("Inserting:"+data);  
  
 }  
  
  
 public boolean isEmpty(){  
  
 return length==0;  
  
  
 }  
  
 public int peek(){  
  
 if(isEmpty()){  
  
 System.*out*.println("Stack is empty");  
 }  
  
 return top.data;  
  
  
 }  
  
 *//remove element from stack* public int pop(){  
 if (isEmpty()){  
 System.*out*.println("Stack is empty");  
 }  
  
  
 int TopData=top.data;  
 top=top.next; *//top will moved to top.next and previously that was top will removed* length--;  
 return TopData;*//now top would be top.next and data stored on it would be returned* }  
  
  
 public static void main(String[] args) {  
 StackInLinkedList stack1=new StackInLinkedList();  
  
 stack1.push(2);  
 stack1.push(3);  
 stack1.push(4);  
 stack1.push(5);  
 System.*out*.println(" top element is:"+stack1.peek());  
  
  
 System.*out*.print("Removed: "+stack1.pop());  
  
 System.*out*.println("After removing top element is:"+stack1.peek());  
  
 if(stack1.isEmpty()){  
  
  
 System.*out*.println("Stack is empty");  
 }  
  
 else {  
  
 System.*out*.println("Stack is not empty");  
 }  
  
}

}

OUTPUT

Sukkur IBA University