**Lab No.9 1)** import java.util.Arrays; import java.util.List;

class Main

{

public static final <T> void swap (T[] a, int i, int j)

{

T t = a[i]; a[i] = a[j];

a[j] = t;

}

public static void main(String[] args)

{

String [] a = {"Hello", "Goodbye"}; swap(a, 0, 1);

System.out.println("a:"+Arrays.toString(a));

Integer [] b = {0, 1}; swap(b, 0, 1);

System.out.println("a:"+Arrays.toString(b));

}

}

**2)** import java.lang.reflect.Array;

class PushException extends Exception

{

private int code;

public PushException(int c)

{

this.code = c;

}

public int getCode()

{ return code;

}

}

class PopException extends Exception

{

private int code;

public PopException(int c)

{

this.code = c;

}

public int getCode()

{ return code;

}

}

class Stack<T>

{

private T item[]; private int top; private int size;

public Stack(Class<T[]> clazz, int length)

{

this.size = length;

this.item = clazz.cast(Array.newInstance(clazz.getComponentType(), length)); this.top = -1;

}

public boolean isEmpty()

{

if(this.top == -1) return (true);

return (false);

}

public boolean isFull(){ if(this.top == this.size -1) return (true);

return (false);

}

public boolean push(T elem) throws PushException

{

if(this.isFull())

{ throw new PushException(1);

}

this.item[++this.top] = elem;

return (true); }

public T pop() throws PopException

{

if(this.isEmpty())

{ throw new PopException(-1);

}

return(this.item[this.top--]);

}

public void display()

{

if(this.isEmpty()) return;

for(int i = 0; i < this.top + 1; i++)

{

System.out.print(this.item[i]);

System.out.print(" "); }

System.out.println("");

}

}

class Student

{

String name; int reg\_no;

Student(String name, int reg\_no)

{ this.name = name;

this.reg\_no = reg\_no;

}

}

class Employee

{

String name; int emp\_no;

Employee(String name, int emp\_no)

{ this.name = name; this.emp\_no = emp\_no;

}

}

class StackTest

{

public static void main(String[] args) {

System.out.println("Demonstrating Generic stack class");

System.out.println("Creating stack object for type Student with size 3"); Stack<Student> s1 = new Stack<Student>(Student[].class, 3);

System.out.println("Creating stack object for type Employee with size 3");

Stack<Employee> s2 = new Stack<Employee>(Employee[].class, 3);

System.out.println("Displaying student stack"); try

{

System.out.println("Pushing elements to student stack"); s1.push(new Student("abcde", 12345)); s1.push(new Student("fghij", 67890)); s1.push(new Student("klmno", 13578));

System.out.println("Displaying student stack"); s1.display();

System.out.println("Pushing elements to employee stack");

s2.push(new Employee("zxcvb", 12568)); s2.push(new Employee("asdfg", 23579)); s2.push(new Employee("qwert", 14795));

System.out.println("Displaying employee stack"); s2.display();

}catch(PushException ex)

{

System.out.println("Caught push exception"); }

}

}

**3)** import java.util.Arrays;

import java.util.List;

class WildcardDemo

{

private static void printlist(List<?> list)

{

System.out.println(list);

}

public static void main(String[] args)

{

List<String> list1= Arrays.asList("abcd", "efgh", "ijkl"); List<Integer> list2=Arrays.asList(10, 20, 30); printlist(list1); printlist(list2);

}