Manoj M Mallya

200905130

Section : C2

Roll no. : 23

LAB – 7 – GENREICS

Lab Exercises :

1. Write a generic method to exchange the positions of two different elements in an array.

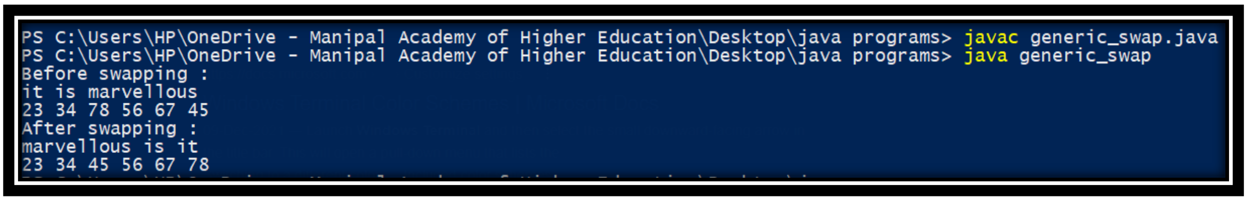
   Code :

```
public class generic_swap

{

    static <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[] = {"it","is","marvellous"};
```

```
System.out.println("Before swapping : ");
for(int i=0;i<a.length;i++)
{
   System.out.print(a[i]+" ");
}
System.out.println();
swap(a,0,2);
Integer b[]={23,34,78,56,67,45};
for(int i=0;i<b.length;i++)
{
   System.out.print(b[i]+" ");
}
swap(b,2,5);
System.out.println("\nAfter swapping :");
for(int i=0;i<a.length;i++)
{
   System.out.print(a[i]+" ");
}
System.out.println();
for(int i=0;i<b.length;i++)
{
   System.out.print(b[i]+" ");
}
```

```
          System.out.println();
      }
  }
```

Output :



```
PS C:\Users\HP\OneDrive - Manipal Academy of Higher Education\Desktop\java programs> javac generic_swap.java
PS C:\Users\HP\OneDrive - Manipal Academy of Higher Education\Desktop\java programs> java generic_swap
Before swapping :
it is marvellous
23 34 78 56 67 45
After swapping :
marvellous is it
23 34 45 56 67 78
```

2. Define a simple generic stack class and show the use of the generic class for two different class types Student and Employee class objects.

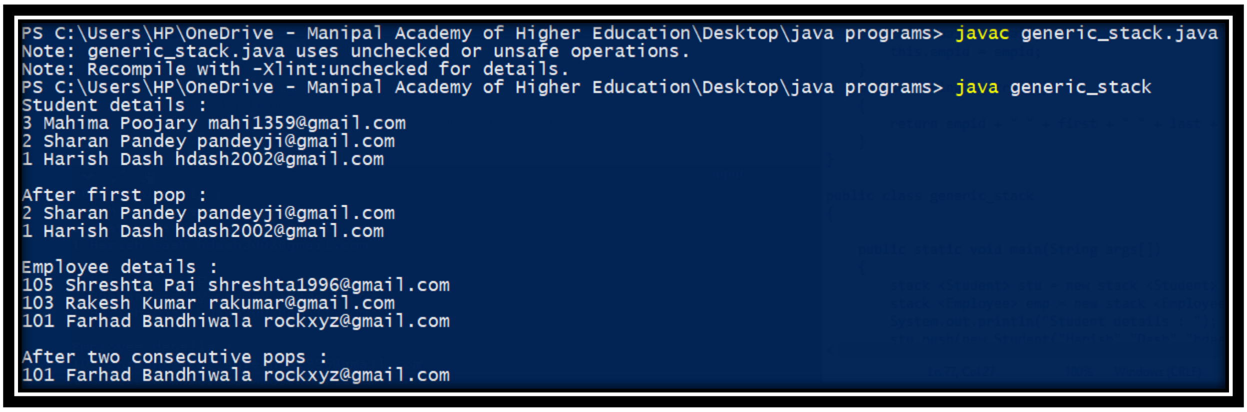
   Code :

```
import java.util.*;


class stack<T>
{
  T[] stk = (T[]) new Object[20];
  static int max = 20;
  int top = -1;


  void push(T ele)
  {
    if(top==max-1)
    {
      System.out.println("Stack is full.");
      return;
```

```
        }
        stk[++top]=ele;
    }

    T pop()
    {
        if (top==-1)
        {
            System.out.println("Stack is empty.");
        }
        return stk[--top];
    }

    void display()
    {
        for(int i=top;i>=0;i--)
        {
            System.out.print(stk[i].toString());
        }
        System.out.println();
    }

}
```

```
class Student
{
    String first;
    String last;
    String email;
    int section;
    public Student(String first,String last,String email,int section)
    {
        this.first = first;
        this.last =  last;
        this.email = email;
        this.section = section;
    }
    public String toString()
    {
        return section + " " + first + " " + last + " " + email + "\n" ;
    }
}

class Employee
{
    String first;
    String last;
    String email;
    int empid;//employee id
    public Employee(String first,String last,String email,int empid)
    {
        this.first = first;
        this.last =  last;
        this.email = email;
        this.empid = empid;
    }
    public String toString()
    {
        return empid + " " + first + " " + last + " " + email + "\n" ;
    }
}

public class generic_stack
{

    public static void main(String args[])
    {
        stack <Student> stu = new stack <Student> ();
        stack <Employee> emp = new stack <Employee> ();
        System.out.println("Student details : ");
        stu.push(new Student("Harish","Dash","hdash2002@gmail.com",1));
        stu.push(new Student("Sharan","Pandey","pandeyji@gmail.com",2));
        stu.push(new Student("Mahima","Poojary","mahi1359@gmail.com",3));
        stu.display();
        stu.pop();
        System.out.println("After first pop : ");
        stu.display();
        System.out.println("Employee details : ");
        emp.push(new
Employee("Farhad","Bandhiwala","rockxyz@gmail.com",101));
        emp.push(new
Employee("Rakesh","Kumar","rakumar@gmail.com",103));
        emp.push(new
Employee("Shreshta","Pai","shreshta1996@gmail.com",105));
        emp.display();
        System.out.println("After two consecutive pops : ");
        emp.pop();
        emp.pop();
        emp.display();
    }
}
```

Output :



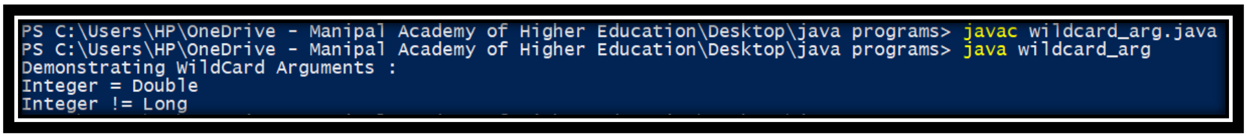
```
PS C:\Users\HP\OneDrive - Manipal Academy of Higher Education\Desktop\java programs> javac generic_stack.java
Note: generic_stack.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
PS C:\Users\HP\OneDrive - Manipal Academy of Higher Education\Desktop\java programs> java generic_stack
Student details :
3 Mahima Poojary mahi1359@gmail.com
2 Sharan Pandey pandeyji@gmail.com
1 Harish Dash hdash2002@gmail.com

After first pop :
2 Sharan Pandey pandeyji@gmail.com
1 Harish Dash hdash2002@gmail.com

Employee details :
105 Shreshta Pai shreshta1996@gmail.com
103 Rakesh Kumar rakumar@gmail.com
101 Farhad Bandhiwala rockxyz@gmail.com

After two consecutive pops :
101 Farhad Bandhiwala rockxyz@gmail.com
```

3. Write a program to demonstrate the use of wildcard arguments.

   Code :

```
class NumFns<T extends Number> {

T num;

NumFns(T n) {

num =n;

}

boolean absEqual (NumFns<?> ob) {

if(Math.abs(num.doubleValue()) == Math.abs(ob.num.doubleValue()))

return true;

return false;

}

}

public class wildcard_arg{

public static void main(String[] args) {

NumFns<Integer> i = new NumFns<Integer> (8);

NumFns<Double> d = new NumFns<Double> (-8.0);
NumFns<Long> l = new NumFns<Long> (6L);

System.out.println("Demonstrating WildCard Arguments : ");

if(i.absEqual(d))

System.out.println("Integer = Double");

else

System.out.println("Integer != Double");

if(i.absEqual(l))

System.out.println("Integer = Long");

else

System.out.println("Integer != Long");

}

}
```

Output :



```
PS C:\Users\HP\OneDrive - Manipal Academy of Higher Education\Desktop\java programs> javac wildcard_arg.java
PS C:\Users\HP\OneDrive - Manipal Academy of Higher Education\Desktop\java programs> java wildcard_arg
Demonstrating WildCard Arguments :
Integer = Double
Integer != Long
```

****