LAB 5:

A local ticket counter maintains a stack of Persons who are standing in line. Each Person has:

- **name (String)**
- **age (int)**
- **gender (enum: MALE, FEMALE, OTHER)**

The counter wants you to implement a Stack of Persons with the following requirements:

1.Person Class

- **Attributes: name, age, gender**
- **Constructor(s) to initialize Person**
- **Method: displayPerson() → prints details of that person**

2. Stack Class

- **A stack that stores Person objects (not just integers).**
- **Two constructors:**
  - **Stack(int size) → creates empty stack of given size**
  - **Stack(Person[] arr) → initializes stack directly from array of Persons**
- **Push Operation (Overloaded):**
  - **push(Person p) → Push single Person**
  - **push(Person p1, Person p2) → Push two Persons together**
- **Pop Operation (Overloaded):**
  - **pop() → Pop one Person**
  - **pop(int n) → Pop n Persons**
- **Display Operation (Overloaded):**
  - **display() → Show full stack (all Persons with details)**
  - **display(int n) → Show only top n Persons**

3. Main Program (Menu Driven)

Allow user to:

1. **Push one Person (enter details: name, age, gender)**
2. **Push two Persons (enter both details)**
3. **Pop one Person**

4. **Pop multiple Persons**
5. **Display all Persons in stack**
6. **Display top n Persons**
7. **Exit**

Github Link: https://github.com/Harshith161/Java-Progs

Code:

```
package stack; enum Gender
{
   MALE,FEMALE,OTHER;
}

package stack; class Person
{
    private String name;
   private int age;
   private Gender gender;
    Person(String name,int age,Gender gender)
   {
      this.name=name;
      this.age=age;
      this.gender=gender;
   }

void displayPerson()
{
   System.out.println("Name is " +name);
   System.out.println("Age is" +age);
   System.out.println("Gender is" +gender);
```

```
 }
}

package stack; class Stack
{
    private Person arr[];
    private int top;
    private int capacity;
    Stack(int size)
   {
        capacity=size;
        top=-1;
        arr = new Person[capacity];
   }

   Stack(Person[] inputArr)
  {
            capacity= inputArr.length;
                arr = new Person[capacity];
                top=-1;
            for(int i=0;i<inputArr.length;i++)
            {
                  arr[++top]=inputArr[i];
            }
}
void push(Person p)
{
       if(top==capacity-1)
      {
             System.out.println("Stack overflow");
```

```
            return;
      }
            arr[++top] = p;
}

void push(Person p1,Person p2)
{
        push(p1);
        push(p2);
}
Person pop()
{       if(top == -1)
     {
          System.out.println("Stack underflow");
              return null;
     }
              return arr[top--];
}

void pop(int n)
{
        for(int i=0;i<n;i++)
       {
          if(top==-1)
          {
               System.out.println("Stack underflow");
      break;
          }
          Person p = pop();
if(p!=null)
```

```
            p.displayPerson();
        }
    }
    void display()
    {
        if(top==-1)
        {
            System.out.println("Stack is empty");
            return;
        }
        for(int i=top;i>=0;i--)
        {
            arr[i].displayPerson();
        }
    }

    void display(int n)
    {
        if(top==-1)
        {
            System.out.println("Stack is empty");
            return;
        }
        for(int i=top;i>top-n && i>=0;i--)
        {
            arr[i].displayPerson();
        }
    }
}
```

package stack; import java.util.*; public class TicketCounterStack

```
{
        public static void main(String[] args)
         {
                Scanner sc = new Scanner(System.in);
                Stack st = null;
                System.out.println("Choose initialization :");
                System.out.println("1. Empty stack with size");
                System.out.println("2. Stack initialized with array of Persons");          int
choice = sc.nextInt();          if(choice == 1)
                {
                        System.out.println("Enter size:");
 int size = sc.nextInt();   st = new Stack(size);
 }
 else if(choice == 2)
 {
                System.out.println("Enter number of Persons:");
                        int n = sc.nextInt();                   Person arr[] = new Person[n];
                for(int i=0;i<n;i++)
                        {
                                System.out.println("Enter the name:");
                                String name = sc.next();
   System.out.println("Enter the age:");      int age = sc.nextInt();
                                System.out.println("Enter gender(MALE/FEMALE/OTHER):");
                        Gender g=Gender.valueOf(sc.next().toUpperCase());
        arr[i] = new Person(name,age,g);
                        }
                        st = new Stack(arr);
                }

                int option;
                do
```

```
{
        System.out.println("\n---Menu---");
        System.out.println("1.Push one Person");
        System.out.println("2.Push two Persons");
        System.out.println("3.Pop one Person");
        System.out.println("4.Pop multiple Persons");
    System.out.println("5.Display all Persons");
    System.out.println("6.Display top n Persons");
    System.out.println("7.Exit");
  System.out.println("Enter your choice:");      option=sc.nextInt();
            switch(option)
    {
    case 1:System.out.println("Enter the name:");
String name_1 = sc.next();                          System.out.println("Enter the age:");
      int age_1 = sc.nextInt();
          System.out.println("Enter gender(MALE/FEMALE/OTHER):");
      Gender g1=Gender.valueOf(sc.next().toUpperCase());
st.push(new Person(name_1,age_1,g1));
        break;
    case 2:System.out.println("Enter first Person's name:");
            String n1 = sc.next();
            System.out.println("Enter the age:");
int a1 = sc.nextInt();
            System.out.println("Enter
gender(MALE/FEMALE/OTHER):");
            Gender g2=Gender.valueOf(sc.next().toUpperCase());
            Person p1=new Person(n1,a1,g2);
                System.out.println("Enter second Person's name:");
            String n2 = sc.next();
            System.out.println("Enter the age:");
int a2 = sc.nextInt();
           System.out.println("Enter
gender(MALE/FEMALE/OTHER):");
```

```
                                        Gender g3=Gender.valueOf(sc.next().toUpperCase());
                        Person p2=new Person(n2,a2,g3);                    st.push(p1,p2);
break;
                    case 3:Person popped = st.pop();
                      if(popped!=null)              popped.displayPerson();
                                        break;                    case 4:System.out.print("Enter
number of Persons to pop:");                                      int n = sc.nextInt();
                        st.pop(n);                              break;                    case
5:st.display();                         break;                    case
6:System.out.print("Enter number of top Persons:");
                                        int topN = sc.nextInt();
st.display(topN);                                       break;
                        case 7:System.out.println("Exiting...");
break;                          default: System.out.println("Invalid choice!");
                        }
                    }
                    while(option!= 7);
            }
}
```

OUTPUT:

Choose initialization :

1. **Empty stack with size**
2. **Stack initialized with array of Persons 2**

Enter number of Persons: 2

Enter the name: DCFG

Enter the age: 20

Enter gender(MALE/FEMALE/OTHER): FEMALE

Enter the name: Harshith

Enter the age: 19

Enter gender(MALE/FEMALE/OTHER): MALE

---Menu---

1.Push one Person

2.Push two Persons

3.Pop one Person

4.Pop multiple Persons

5.Display all Persons

6.Display top n Persons

7.Exit

Enter your choice:

5

Name is Harshith

Age is19

Gender is MALE

Name is

Age is19 DCFG

Gender isFEMALE

---Menu---

1.Push one Person

2.Push two Persons

3.Pop one Person

4.Pop multiple Persons

5.Display all Persons

6.Display top n Persons

7.Exit

Enter your choice: 3

Name is Harshith

Age is19

Gender isMALE

---Menu---

1.Push one Person

2.Push two Persons

3.Pop one Person

4.Pop multiple Persons

5.Display all Persons

6.Display top n Persons

7.Exit

Enter your choice:

7

Exiting...