Code:

// ENUM for Gender

enum Gender {

MALE, FEMALE;

}

// Person Class

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: " + this.name + ", Age: " + this.age + ", Gender: " + this.gender);

}

}

// Stack Class

class Stack {

private Person arr[];

private int top;

private int capacity;

Stack(int size) {

this.capacity = size;

this.arr = new Person[size];

this.top = -1;

}

Stack(Person[] inputArr) {

this.capacity = inputArr.length;

this.arr = new Person[capacity];

this.top = -1;

for (Person p : inputArr) {

push(p);

}

}

// Push single Person

void push(Person p) {

if (isFull()) {

System.out.println("Stack Overflow. Cannot push " + p.name);

return;

}

arr[++top] = p;

System.out.println(p.name + " pushed to stack.");

}

CS2002-1 (Object Oriented Programming) 4 |

Page

// Overloaded Push

void push(Person p1, Person p2) {

push(p1);

push(p2);

}

// Pop single Person

Person pop() {

if (isEmpty()) {

System.out.println("Stack Underflow.");

return null;

}

Person poppedPerson = arr[top--];

System.out.println(poppedPerson.name + " popped from stack.");

return poppedPerson;

}

// Overloaded Pop

void pop(int n) {

if (top - n + 1 < 0) {

System.out.println("Stack Underflow. Cannot pop " + n + " persons.");

return;

}

for (int i = 0; i < n; i++) {

pop();

}

}

// Display all Persons

void display() {

if (isEmpty()) {

System.out.println("Stack is empty.");

return;

}

System.out.println("All Persons in Stack");

for (int i = top; i >= 0; i--) {

arr[i].displayPerson();

}

}

// Overloaded Display

void display(int n) {

if (isEmpty()) {

System.out.println("Stack is empty.");

return;

}

if (n > top + 1) {

System.out.println("Cannot display " + n + " persons. Stack only contains " + (top + 1) + "

persons.");

n = top + 1;

}

System.out.println("--- Top " + n + " Persons in Stack ");

for (int i = top; i > top - n; i--) {

arr[i].displayPerson();

}

}

boolean isEmpty() {

return top == -1;

}

CS2002-1 (Object Oriented Programming) 6 |

Page

boolean isFull() {

return top == capacity - 1;

}

public int size() {

return top + 1;

}

}

//Main Class

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();

sc.nextLine();

if (choice == 1) {

System.out.print("Enter size: ");

int size = sc.nextInt();

st = new Stack(size);

} else if (choice == 2) {

CS2002-1 (Object Oriented Programming) 7 |

Page

System.out.print("Enter number of Persons: ");

int n = sc.nextInt();

sc.nextLine();

Person arr[] = new Person[n];

for (int i = 0; i < n; i++) {

System.out.println("Enter details for Person " + (i + 1) + ":");

System.out.print("Name: ");

String name = sc.nextLine();

System.out.print("Age: ");

int age = sc.nextInt();

sc.nextLine();

System.out.print("Gender (MALE, FEMALE): ");

Gender gender = Gender.valueOf(sc.nextLine().toUpperCase());

arr[i] = new Person(name, age, gender);

}

st = new Stack(arr);

} else {

System.out.println("Invalid choice. Exiting...");

sc.close();

return;

}

int option;

do {

System.out.println("\nMenu");

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.print("Enter your choice: ");

option = sc.nextInt();

sc.nextLine();

switch (option) {

case 1:

System.out.print("Name: ");

String name1 = sc.nextLine();

System.out.print("Age: ");

int age1 = sc.nextInt();

sc.nextLine();

System.out.print("Gender (MALE, FEMALE): ");

Gender gender1 = Gender.valueOf(sc.nextLine().toUpperCase());

st.push(new Person(name1, age1, gender1));

break;

case 2:

System.out.println("Enter details for Person 1:");

System.out.print("Name: ");

String nameA = sc.nextLine();

System.out.print("Age: ");

int ageA = sc.nextInt();

sc.nextLine();

System.out.print("Gender (MALE, FEMALE): ");

Gender genderA = Gender.valueOf(sc.nextLine().toUpperCase());

System.out.println("Enter details for Person 2:");

System.out.print("Name: ");

String nameB = sc.nextLine();

System.out.print("Age: ");

int ageB = sc.nextInt();

sc.nextLine();

System.out.print("Gender (MALE, FEMALE): ");

Gender genderB = Gender.valueOf(sc.nextLine().toUpperCase());

st.push(new Person(nameA, ageA, genderA), new Person(nameB, ageB,

genderB));

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);

sc.close();

}

}

OUTPUT:

Choose initialization:

1. Empty stack with size

2. Stack initialized with array of Persons

1

Enter size: 5

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: 1

Name:chandan

Age: 25

Gender (MALE, FEMALE): FEMALE

Chandan pushed to stack.

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: 2

Enter details for Person 1:

Name: chirag

Age: 48

Gender (MALE, FEMALE): MALE

Enter details for Person 2:

Name: darshan

Age: 22

Gender (MALE, FEMALE): MALE

chirag pushed to stack.

darshan pushed to stack.

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

chandan popped from stack.

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

chandan popped from stack.

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

chandan popped from stack.

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

Stack is empty.

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...