**Problem Statement 1: Java 8- Lambda Expressions**

**1.Create an interface Instrument which is having the abstract function play. Create three implementations from Instrument which is Piano, Flute, Guitar using Lambda Expressions. Override the play method inside all three classes printing a message as follows,**

**“Piano is playing tan tan tan tan” for Piano class**

**“Flute is playing toot toot toot toot” for Flute class**

**Note: You must not allow the user to declare an object of Instrument class.**

**• Create an array of 10 Instruments.**

**• Assign different type of instrument-to-Instrument reference.**

**• Check for the polymorphic behavior of play method.**

**• Use the instanceof operator to print that which object stored at which index of instrument array.**

**package** day6assignment;

**public** **class** Instruments {

**interface** Instrument {

**void** play();

}

**static** **class** Piano **implements** Instrument {

**public** **void** play() {

System.***out***.println("Piano is playing tan tan tan tan");

}

}

**static** **class** Flute **implements** Instrument {

**public** **void** play() {

System.***out***.println("Flute is playing toot toot toot toot");

}

}

**static** **class** Guitar **implements** Instrument {

**public** **void** play() {

System.***out***.println("Guitar is playing strum strum strum strum");

}

}

**public** **static** **void** main(String[] args) {

Instrument[] instruments = **new** Instrument[10];

instruments[0] = **new** Piano();

instruments[1] = **new** Flute();

instruments[2] = **new** Guitar();

instruments[3] = **new** Piano();

instruments[4] = **new** Flute();

instruments[5] = **new** Guitar();

instruments[6] = **new** Piano();

instruments[7] = **new** Flute();

instruments[8] = **new** Guitar();

instruments[9] = **new** Piano();

**for** (**int** i = 0; i < instruments.length; i++) {

System.***out***.print("Instrument at index " + i + ": ");

**if** (instruments[i] **instanceof** Piano) {

System.***out***.print("Piano -> ");

} **else** **if** (instruments[i] **instanceof** Flute) {

System.***out***.print("Flute -> ");

} **else** **if** (instruments[i] **instanceof** Guitar) {

System.***out***.print("Guitar -> ");

}

instruments[i].play();

}

}

}

**Output:**

Instrument at index 0: Piano -> Piano is playing tan tan tan tan

Instrument at index 1: Flute -> Flute is playing toot toot toot toot

Instrument at index 2: Guitar -> Guitar is playing strum strum strum strum

Instrument at index 3: Piano -> Piano is playing tan tan tan tan

Instrument at index 4: Flute -> Flute is playing toot toot toot toot

Instrument at index 5: Guitar -> Guitar is playing strum strum strum strum

Instrument at index 6: Piano -> Piano is playing tan tan tan tan

Instrument at index 7: Flute -> Flute is playing toot toot toot toot

Instrument at index 8: Guitar -> Guitar is playing strum strum strum strum

Instrument at index 9: Piano -> Piano is playing tan tan tan tan

**Problem Statement 2: New Date-Time API in Java 8**

**2.HealthBox is an online Life Coaching application that helps its users to sign up and log in to seek the guidance of famous Life Coaches across the world from different expertise in one place. Users can book an appointment for Life Coach based on specialty. They can also see upcoming appointments and can reschedule or cancel the appointments.**

**Provide a menu for the user to choose from the below options and perform accordingly:**

**1. Schedule an appointment where customer can enter date, time, and zone to which they belong.**

**2. Print appointment details only if it is booked if not display appropriate message.**

**3. Reschedule an appointment.**

**4. Get Reminder where customer can check for the schedule one day prior to the appointment.**

**5. We can cancel the appointments before the scheduled date only if not display appropriate message.**

**6. Exit.**

package day6assignment;

import java.util.Scanner;

import java.time.LocalDate;

import java.time.LocalTime;

import java.time.format.DateTimeFormatter;

public class HealthBox {

private static LocalDate *appointmentDate* = null;

private static LocalTime *appointmentTime* = null;

private static String *appointmentZone* = "";

private static boolean *isAppointmentBooked* = false;

private static final DateTimeFormatter *dateFormatter* = DateTimeFormatter.*ofPattern*("dd-MM-yyyy");

private static final DateTimeFormatter *timeFormatter* = DateTimeFormatter.*ofPattern*("HH:mm");

public static void main(String[] args) {

Scanner scanner = new Scanner(System.*in*);

while (true) {

System.*out*.println("\nHealthBox Life Coaching Application");

System.*out*.println("1. Schedule an appointment");

System.*out*.println("2. Print appointment details");

System.*out*.println("3. Reschedule an appointment");

System.*out*.println("4. Get Reminder");

System.*out*.println("5. Cancel the appointment");

System.*out*.println("6. Exit");

System.*out*.println("===================================");

System.*out*.print("Enter your choice: ");

System.*out*.println("\n===================================");

int choice = scanner.nextInt();

scanner.nextLine();

switch (choice) {

case 1:

*scheduleAppointment*(scanner);

break;

case 2:

*printAppointmentDetails*();

break;

case 3:

*rescheduleAppointment*(scanner);

break;

case 4:

*getReminder*();

break;

case 5:

*cancelAppointment*(scanner);

break;

case 6:

System.*out*.println("Exiting the application. Thank you!");

scanner.close();

return;

default:

System.*out*.println("Invalid choice. Please try again.");

}

}

}

private static void scheduleAppointment(Scanner scanner) {

if (*isAppointmentBooked*) {

System.*out*.println("You already have a booked appointment. Reschedule or cancel it first.");

return;

}

System.*out*.print("Enter appointment date (DD-MM-YYYY): ");

String dateInput = scanner.nextLine();

System.*out*.print("Enter appointment time (HH:MM): ");

String timeInput = scanner.nextLine();

System.*out*.print("Enter your time zone: ");

*appointmentZone* = scanner.nextLine();

try {

*appointmentDate* = LocalDate.*parse*(dateInput, *dateFormatter*);

*appointmentTime* = LocalTime.*parse*(timeInput, *timeFormatter*);

*isAppointmentBooked* = true;

System.*out*.println("Appointment successfully booked!");

} catch (Exception e) {

System.*out*.println("Invalid date or time format. Please try again.");

}

}

private static void printAppointmentDetails() {

if (*isAppointmentBooked*) {

System.*out*.println("\nAppointment Details:");

System.*out*.println("Date: " + *appointmentDate*.format(*dateFormatter*));

System.*out*.println("Time: " + *appointmentTime*.format(*timeFormatter*));

System.*out*.println("Time Zone: " + *appointmentZone*);

} else {

System.*out*.println("No appointment is scheduled yet.");

}

}

private static void rescheduleAppointment(Scanner scanner) {

if (!*isAppointmentBooked*) {

System.*out*.println("You do not have an appointment to reschedule.");

return;

}

System.*out*.println("You already have a booked appointment. Rescheduling...");

System.*out*.print("Enter new appointment date (DD-MM-YYYY): ");

String newDateInput = scanner.nextLine();

System.*out*.print("Enter new appointment time (HH:MM): ");

String newTimeInput = scanner.nextLine();

System.*out*.print("Enter your time zone: ");

*appointmentZone* = scanner.nextLine();

try {

*appointmentDate* = LocalDate.*parse*(newDateInput, *dateFormatter*);

*appointmentTime* = LocalTime.*parse*(newTimeInput, *timeFormatter*);

System.*out*.println("Appointment successfully rescheduled!");

} catch (Exception e) {

System.*out*.println("Invalid date or time format. Please try again.");

}

}

private static void getReminder() {

if (*isAppointmentBooked*) {

System.*out*.println("\nReminder: Your appointment is scheduled for " + *appointmentDate*.format(*dateFormatter*) + " at " + *appointmentTime*.format(*timeFormatter*) + " (" + *appointmentZone* + ").");

} else {

System.*out*.println("No appointment is scheduled to remind you about.");

}

}

private static void cancelAppointment(Scanner scanner) {

if (!*isAppointmentBooked*) {

System.*out*.println("You do not have an appointment to cancel.");

return;

}

System.*out*.print("Enter appointment date (DD-MM-YYYY) to cancel: ");

String cancelDate = scanner.nextLine();

try {

LocalDate cancelDateParsed = LocalDate.*parse*(cancelDate, *dateFormatter*);

if (cancelDateParsed.equals(*appointmentDate*)) {

*isAppointmentBooked* = false;

System.*out*.println("Your appointment has been successfully canceled.");

} else {

System.*out*.println("You can only cancel the appointment if it's before the scheduled date.");

}

} catch (Exception e) {

System.*out*.println("Invalid date format. Please try again.");

}

}

}

**Output:**

HealthBox Life Coaching Application

1. Schedule an appointment

2. Print appointment details

3. Reschedule an appointment

4. Get Reminder

5. Cancel the appointment

6. Exit

===================================

Enter your choice:

===================================

**1**

**Enter appointment date (DD-MM-YYYY): 05-01-2025**

**Enter appointment time (HH:MM): 10:30**

**Enter your time zone: bangalore**

**Appointment successfully booked!**

HealthBox Life Coaching Application

1. Schedule an appointment

2. Print appointment details

3. Reschedule an appointment

4. Get Reminder

5. Cancel the appointment

6. Exit

===================================

Enter your choice:

===================================

**2**

**Appointment Details:**

**Date: 05-01-2025**

**Time: 10:30**

**Time Zone: bangalore**

HealthBox Life Coaching Application

1. Schedule an appointment

2. Print appointment details

3. Reschedule an appointment

4. Get Reminder

5. Cancel the appointment

6. Exit

===================================

Enter your choice:

===================================

**3**

**You already have a booked appointment. Rescheduling...**

**Enter new appointment date (DD-MM-YYYY): 08-01-2025**

**Enter new appointment time (HH:MM): 09:00 am**

**Enter your time zone: Bangalore**

Invalid date or time format. Please try again.

HealthBox Life Coaching Application

1. Schedule an appointment

2. Print appointment details

3. Reschedule an appointment

4. Get Reminder

5. Cancel the appointment

6. Exit

===================================

Enter your choice:

===================================

**4**

**Reminder: Your appointment is scheduled for 08-01-2025 at 10:30 (Bangalore).**

HealthBox Life Coaching Application

1. Schedule an appointment

2. Print appointment details

3. Reschedule an appointment

4. Get Reminder

5. Cancel the appointment

6. Exit

===================================

Enter your choice:

===================================

**5**

**Enter appointment date (DD-MM-YYYY) to cancel: 08-01-2025**

**Your appointment has been successfully canceled.**

HealthBox Life Coaching Application

1. Schedule an appointment

2. Print appointment details

3. Reschedule an appointment

4. Get Reminder

5. Cancel the appointment

6. Exit

===================================

Enter your choice:

===================================

**6**

**Exiting the application. Thank you!**

**Problem Statement 3: Design the highly general and reusable code with Generic classes.**

**3.1 Astound has recently opened its internet services in India. The company wants users to register for their internet services. Presently, there are two kinds of users, i.e., employees and students.**

**However, the company has establishedthe following criteria for user authentication:**

**• Firstly, for both the type of users, Name and two Phone No (one alternate phone number) is compulsory.**

**• Also, the user should have passport. If user is not having a passport, he/she can provide anyone of the following combinations.**

**I. License number and pan card.**

**II. Voter Id and License number.Adding on criteriamandatory for an Employee:**

**• Provide employee Id.**

**Design a generic class of name Register having methods to generate registration id and display based on the type of user.**

package day6assignment;

import java.util.Random;

import java.util.Scanner;

class Employee{

String name,passportno,PANcardno;

long[] phoneno=new long[2];

int licenseno,voterID,empID;

public Employee(String name,long[] phoneno,String passportno,int empID) {

this.name=name;

this.phoneno=phoneno;

this.passportno=passportno;

this.empID=empID;

}

public Employee(String name,long[] phoneno,String PANcardno,int licenseno,int empID) {

this.name=name;

this.phoneno=phoneno;

this.PANcardno=PANcardno;

this.licenseno=licenseno;

this.empID=empID;

}

public Employee(String name,long[] phoneno,int voterID,int licenseno,int empID) {

this.name=name;

this.phoneno=phoneno;

this.voterID=voterID;

this.licenseno=licenseno;

this.empID=empID;

}

public void display() {

System.*out*.println("Employee Name: " + name);

System.*out*.println("Phone Numbers: " + phoneno[0] + ", " + phoneno[1]);

if (passportno != null && !passportno.isEmpty()) {

System.*out*.println("Passport No: " + passportno);

}

if (licenseno != 0) {

System.*out*.println("License No: " + licenseno);

}

if (PANcardno != null && !PANcardno.isEmpty()) {

System.*out*.println("PAN Card No: " + PANcardno);

}

if (voterID != 0) {

System.*out*.println("Voter ID: " + voterID);

}

System.*out*.println("Employee ID: " + empID);

}

}

class Student{

String name,passportno,PANcardno;

long[] phoneno=new long[2];

int licenseno,voterID;

public Student(String name,long[] phoneno,String passportno) {

this.name=name;

this.phoneno=phoneno;

this.passportno=passportno;

}

public Student(String name,long[] phoneno,String PANcardno,int licenseno) {

this.name=name;

this.phoneno=phoneno;

this.PANcardno=PANcardno;

this.licenseno=licenseno;

}

public Student(String name,long[] phoneno,int voterID,int licenseno) {

this.name=name;

this.phoneno=phoneno;

this.voterID=voterID;

this.licenseno=licenseno;

}

public void display() {

System.*out*.println("Student Name: " + name);

if (passportno != null && !passportno.isEmpty()) {

System.*out*.println("Passport No: " + passportno);

}

if (licenseno != 0) {

System.*out*.println("License No: " + licenseno);

}

if (PANcardno != null && !PANcardno.isEmpty()) {

System.*out*.println("PAN Card No: " + PANcardno);

}

if (voterID != 0) {

System.*out*.println("Voter ID: " + voterID);

}

}

}

class Register<T> {

String registerId;

public String generateRegisterId(int n) {

String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";

Random random = new Random();

StringBuilder registerIdBuilder = new StringBuilder(n);

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

int index = random.nextInt(characters.length());

registerIdBuilder.append(characters.charAt(index));

}

registerId = registerIdBuilder.toString();

return registerId;

}

public void display(T obj) {

System.*out*.println("Registration ID: " + registerId);

if (obj instanceof Employee) {

((Employee) obj).display();

} else if (obj instanceof Student) {

((Student) obj).display();

}

}

}

public class Tester {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.*in*);

System.*out*.print("Enter user type (Employee/Student): ");

String userType = scanner.nextLine();

if (userType.equalsIgnoreCase("Employee")) {

System.*out*.print("Enter Employee Name: ");

String name = scanner.nextLine();

System.*out*.print("Enter phone numbers (space-separated): ");

long phone1 = scanner.nextLong(), phone2 = scanner.nextLong();

long[] phoneno = {phone1, phone2};

System.*out*.print("Enter Employee ID: ");

int empID = scanner.nextInt();

scanner.nextLine();

System.*out*.print("Enter Passport No: ");

String passportno = scanner.nextLine();

if (!passportno.isEmpty()) {

Register<Employee> register = new Register<>();

Employee employee = new Employee(name, phoneno, passportno, empID);

register.generateRegisterId(8);

System.*out*.println("============Details of employee:============\n");

System.*out*.println("Hurray!! you have availed a discount of 10% ");

register.display(employee);

return;

}

System.*out*.print("Enter License No: ");

Integer licenseno = scanner.nextInt();

scanner.nextLine();

System.*out*.print("Enter PAN Card No: ");

String PANcardno = scanner.nextLine();

if (!PANcardno.isEmpty()) {

Register<Employee> register = new Register<>();

Employee employee = new Employee(name, phoneno, PANcardno, licenseno, empID);

register.generateRegisterId(8);

System.*out*.println("============Details of employee:============\n");

System.*out*.println("Hurray!! you have availed a discount of 10% ");

register.display(employee);

return;

}

System.*out*.print("Enter VoterID: ");

Integer voterID = scanner.nextInt();

Register<Employee> register = new Register<>();

Employee employee = new Employee(name, phoneno, voterID, licenseno, empID);

register.generateRegisterId(8);

System.*out*.println("============Details of employee:============\n");

System.*out*.println("Hurray!! you have availed a discount of 10% ");

register.display(employee);

}else if (userType.equalsIgnoreCase("Student")) {

System.*out*.print("Enter Student Name: ");

String name = scanner.nextLine();

System.*out*.print("Enter phone numbers (space-separated): ");

long phone1 = scanner.nextLong(), phone2 = scanner.nextLong();

long[] phoneno = {phone1, phone2};

scanner.nextLine();

System.*out*.print("Enter Passport No: ");

String passportno = scanner.nextLine();

if (!passportno.isEmpty()) {

Register<Student> register = new Register<>();

Student student = new Student(name, phoneno, passportno);

register.generateRegisterId(8);

System.*out*.println("============Details of student:============\n");

System.*out*.println("Hurray!! you have availed a discount of 22% ");

register.display(student);

return;

}

System.*out*.print("Enter License No: ");

Integer licenseno = scanner.nextInt();

scanner.nextLine();

System.*out*.print("Enter PAN Card No: ");

String PANcardno = scanner.nextLine();

if (!PANcardno.isEmpty()) {

Register<Student> register = new Register<>();

Student student = new Student(name, phoneno, PANcardno, licenseno);

register.generateRegisterId(8);

System.*out*.println("============Details of student:============\n");

System.*out*.println("Hurray!! you have availed a discount of 22% ");

register.display(student);

return;

}

System.*out*.print("Enter VoterID: ");

Integer voterID = scanner.nextInt();

Register<Student> register = new Register<>();

Student student = new Student(name, phoneno, voterID, licenseno);

register.generateRegisterId(8);

System.*out*.println("============Details of student:============\n");

System.*out*.println("Hurray!! you have availed a discount of 22% ");

register.display(student);

}else {

System.*out*.println("Invalid user type.");

}

}

}

**Output1:**

Enter user type (Employee/Student): employee

Enter Employee Name: gaurav

Enter phone numbers (space-separated): 1236547890 0123654789

Enter Employee ID: 1101

Enter Passport No: 4856YH

**============Details of employee:============**

**Hurray!! you have availed a discount of 10%**

**Registration ID: P0EWJU4H**

**Employee Name: gaurav**

**Phone Numbers: 1236547890, 123654789**

**Passport No: 4856YH**

**Employee ID: 1101**

**Output2:**

Enter user type (Employee/Student): employee

Enter Employee Name: sai

Enter phone numbers (space-separated): 9874563210 4569871230

Enter Employee ID: 22001

Enter Passport No:

Enter License No: 2365

Enter PAN Card No: 5415466YGF

**============Details of employee:============**

**Hurray!! you have availed a discount of 10%**

**Registration ID: DVIDWIST**

**Employee Name: sai**

**Phone Numbers: 9874563210, 4569871230**

**License No: 2365**

**PAN Card No: 5415466YGF**

**Employee ID: 22001**

**Output3:**

Enter user type (Employee/Student): employee

Enter Employee Name: rani

Enter phone numbers (space-separated): 2587410369 3698520147

Enter Employee ID: 3002

Enter Passport No:

Enter License No: 56564

Enter PAN Card No:

Enter VoterID: 5685

**============Details of employee:============**

**Hurray!! you have availed a discount of 10%**

**Registration ID: E9LJPF63**

**Employee Name: rani**

**Phone Numbers: 2587410369, 3698520147**

**License No: 56564**

**Voter ID: 5685**

**Employee ID: 3002**

**Output4:**

Enter user type (Employee/Student): student

Enter Student Name: pranav

Enter phone numbers (space-separated): 1236547890 9874563210

Enter Passport No: 5454TYDG

**============Details of student:============**

**Hurray!! you have availed a discount of 22%**

**Registration ID: CKDPBZ8X**

**Student Name: pranav**

**Passport No: 5454TYDG**

**Output5:**

Enter user type (Employee/Student): student

Enter Student Name: mohan

Enter phone numbers (space-separated): 2580147963 3698552014

Enter Passport No:

Enter License No: 5487

Enter PAN Card No: gfd5353

**============Details of student:============**

**Hurray!! you have availed a discount of 22%**

**Registration ID: JHFLWM44**

**Student Name: mohan**

**License No: 5487**

**PAN Card No: gfd5353**

**Output6:**

Enter user type (Employee/Student): student

Enter Student Name: bhanu

Enter phone numbers (space-separated): 0258741369 4560123987

Enter Passport No:

Enter License No: 4546

Enter PAN Card No:

Enter VoterID: 1254

**============Details of student:============**

**Hurray!! you have availed a discount of 22%**

**Registration ID: 2YYN4MVM**

**Student Name: bhanu**

**License No: 4546**

**Voter ID: 1254**