

CS1083

Assignment #0

Daniyal Khan
3765942

Source Code For Attendee Class:

```
/**
This is class represents a Attendee
@author Daniyal Khan 3765942
*/
public class Attendee {
    private int daysAttended;
    private int numOfEventsVisited;

    private Event[] events;
    private int eventsAttended; // companion variable

    public Attendee(int daysAttended) {
        events = new Event[5];
        eventsAttended = 0;
        numOfEventsVisited = 0;
        this.daysAttended = daysAttended;
    }

    public void addEvent(Event event) {
        if(eventsAttended >= 5) {
            Event[] moreEvents = new Event[events.length+1];
            System.arraycopy(events, 0, moreEvents, 0,
events.length);
            events = moreEvents;
        }
        events[eventsAttended] = event;
        numOfEventsVisited++;
        eventsAttended++;
    }

    public boolean removeEvent(Event eventToBeRemoved) {
        boolean removed = false;
        for (int i = 0; i < eventsAttended; i++) {
            if (events[i].getID() == eventToBeRemoved.getID()) {
                for (int j = i; j < eventsAttended-1; j++) {
                    events[j] = events[j+1];
                }
                events[eventsAttended-1] = null;
                eventsAttended--;
                removed = true;
            }
        }
    }
}
```

```

        }
    }
    return removed;
}

public Event[] listOfEvents() {
    return events;
}

public double costOfTicket() {
    double cost = 189.99;
    if (eventsAttended > 4) {
        cost = 279.99;
    }
    return cost;
}

public String textuallistOfEvents() {
    String allEvents = "";
    for (int i = 0; i < eventsAttended; i++) {
        allEvents += events[i].getID() + "    " +
events[i].getArtistName() + "\n";
    }
    return allEvents;
}
}

```



Source code for Event:

```

public class Event{
    private String artistName;
    private int id;
    private static int nextID = 6000;
}

```

```

    public Event(String nameIn){
        artistName = nameIn;
        id = nextID;
        nextID++;
    }

    public String getArtistName(){
        return artistName;
    }

    public int getID(){
        return id; }
}

```

Source Code for VIP attendee:

```

/**
This class represents a VIP attendee which is a subclass of
Attendee
@author Daniyal Khan 3765942
*/
public class VIPAttendee extends Attendee {
    private boolean group;
    private final double DEFAULTFEE;

    public VIPAttendee(boolean group) {
        super(4); // VIPAttendee attend all 4 days
        this.group = group;
        DEFAULTFEE = 209.99;
    }

    public boolean attendingWithGroup() {
        return group;
    }

    public void updateGroup() {
        group = !group;
    }

    public double costOfTicket() {

```

```

double cost = DEFAULTFEE;
Event events[] = super.listOfEvents();
String[] artists = new String[events.length];
int uniqueArtistCount = 0;

for (int i = 0; i < events.length; i++) {
    artists[i] = events[i].getArtistName();
}

for (int j = 0; j < artists.length-1; j++) {
    if (artists[j] != artists[j+1]) {
        uniqueArtistCount++;
    }
}

if(uniqueArtistCount >= 1) { // if attendee is attending
different artist events
    cost = super.costOfTicket();
}

return cost;
}
}

```



Source Code for Driver:

```

/**
This is a driver class for Attendee and VIPAttendee

```

@author *Daniyal Khan 3765942*

**/*

```
public class Driver {
    public static void main(String[] args) {
        // ALL EVENTS
        Event event1 = new Event("Billy Currington");
        Event event2 = new Event("Luke Bryan");
        Event event3 = new Event("High Valley");
        Event event4 = new Event("Connor Price");
        Event event5 = new Event("Timberlake");
        Event event6 = new Event("Marianas Trench");
        Event event7 = new Event("Tim McGraw");
        Event event8 = new Event("Deric Ruttan");
        Event event9 = new Event("Billy Currington");
        Event event10 = new Event("High Valley");
        Attendee attendee1 = new Attendee(2);
        Attendee attendee2 = new Attendee(3);
        VIPAttendee VIPattendee1 = new VIPAttendee(false);
        VIPAttendee VIPattendee2 = new VIPAttendee(true);

        // TEST CASE 1
        System.out.println("");
        System.out.println("-----");
        System.out.println("Test Case 1");
        System.out.println("");
        attendee1.addEvent(event1); // adding events
        attendee1.addEvent(event5);
        attendee1.addEvent(event4);
        attendee1.addEvent(event3);
        attendee1.addEvent(event2);
        attendee1.addEvent(event7);
        attendee1.addEvent(event6); // adding more than 5 events
        sequentially causing creation of a new array

        attendee1.removeEvent(event7); // removing an event
        attendee1.removeEvent(event8); // removing an event that
        does not exist
        System.out.println(attendee1.textuallistOfEvents());
        System.out.println("Total Price: " +
attendee1.costOfTicket());

        // TEST CASE 2
        System.out.println("");
        System.out.println("-----");
```

```

        System.out.println("Test Case 2");
        System.out.println("");
        attendee2.addEvent(event1); // Events added and dropped,
where the current number of events attended
        attendee2.addEvent(event5); // never exceeds 5, but the
total added + dropped exceeds 5.
        attendee2.addEvent(event4);
        attendee2.addEvent(event3);
        attendee2.addEvent(event2);

        attendee2.removeEvent(event5);
        attendee2.addEvent(event7);
        attendee2.removeEvent(event2);
        attendee2.addEvent(event6);
        attendee2.removeEvent(event1);

        System.out.println(attendee2.textuallistOfEvents());
        System.out.println("Total Price: " +
attendee2.costOfTicket());

```

```

// TEST CASE 3
System.out.println("");
System.out.println("-----");
System.out.println("Test Case 3");
System.out.println("");
VIPattendee1.addEvent(event9);
VIPattendee1.addEvent(event1);
VIPattendee1.addEvent(event9);
VIPattendee1.addEvent(event1);
VIPattendee1.addEvent(event9);

        System.out.println(VIPattendee1.textuallistOfEvents());
// VIP Attendee attending same artist events
        System.out.println("Joining with group? " +
VIPattendee2.attendingWithGroup());
        System.out.println("Total Price: " +
VIPattendee1.costOfTicket());

```

```

// TEST CASE 4
System.out.println("");
System.out.println("-----");
System.out.println("Test Case 4");
System.out.println("");
VIPattendee2.addEvent(event8);

```

```
VIPattendee2.addEvent(event2);
VIPattendee2.addEvent(event8);
VIPattendee2.addEvent(event5);
VIPattendee2.addEvent(event4);

    System.out.println(VIPattendee2.textuallistOfEvents());
// VIP Attendee attending different artist events
    System.out.println("Joining with group? " +
VIPattendee2.attendingWithGroup());
    System.out.println("Total Price: " +
VIPattendee2.costOfTicket());
    }
}
```


Output:

```
~/OneDrive - University of New Brunswick/CS-XXXX/CS1083/Assignments/as0 main +4 !4 ?7 04:57:21 pm
java Driver

-----
Test Case 1

6000 Billy Currington
6004 Timberlake
6003 Connor Price
6002 High Valley
6001 Luke Bryan
6005 Marianas Trench

Total Price: 279.99

-----
Test Case 2

6003 Connor Price
6002 High Valley
6006 Tim McGraw
6005 Marianas Trench

Total Price: 189.99

-----
```

Test case 1 demonstrates the **adding of events** to an attendee object, **adding more than 5 events sequentially** causing creation of a new array, **removing an event** and removing an **event that does not exist**.

In **Test case 2** the Events are added and dropped, where the current number of events attended never exceeds 5, but the total added + dropped exceeds 5.

```
-----
Test Case 3
6008  Billy Currington
6000  Billy Currington
6008  Billy Currington
6000  Billy Currington
6008  Billy Currington

Joining with group? true
Total Price: 209.99
-----

Test Case 4
6007  Deric Ruttan
6001  Luke Bryan
6007  Deric Ruttan
6004  Timberlake
6003  Connor Price

Joining with group? true
Total Price: 279.99

~/0/CS-XXXX/CS1083/Assignments/As0  main !11  12:36:19 pm
```

Test case 3 demonstrates the adding of events by the same artists only and checking the VIP price for it. Also checks if they are coming with a group.

Test case 4 demonstrates the VIP attendee attending events with different artists. Also checks if they are coming with a group.