

# Object Oriented Programming

## Lab 07 – Abstraction and Inheritance

16<sup>th</sup> March 2023

### Question:

A set of classes is used to handle the different ticket types for a theatre. All tickets have a unique serial number that is assigned when the ticket is constructed and a price. There are many types of tickets.

a. Implement the Ticket abstract class. This class should store a serial number as its private data. Provide an appropriate abstract method to get the price of the ticket, provide a method that returns the serial number, and provide an implementation of toString that prints the serial number and price information. The Ticket class must provide a constructor to initialise the serial number. To do so, use the following strategy: maintain a static ArrayList<Integer> representing previously assigned serial numbers. Repeatedly generate a new serial number using a random number generator until you obtain a serial number not already assigned.

Ticket type	Description	Sample toString Output
Ticket	This is an abstract class representing all tickets	
FixedPriceTicket	This is an abstract class representing tickets that are always the same price. The constructor accepts the price as the parameter.	
ComplimentaryTicket	These tickets are free (thus FixedPrice).	SN: 273, \$0
WalkupTicket	These tickets are purchased on the day of the event for \$50 (thus FixedPrice).	SN: 314, \$50
AdvanceTicket	Tickets purchased ten or more days in advance cost \$30. Tickets purchased fewer than ten days in advance cost \$40.	SN: 612, \$40
StudentAdvanceTicket	These are AdvanceTickets that cost half of what an AdvanceTicket would normally cost.	SN: 59, \$15  (student)

b. Implement the FixedPriceTicket class. The constructor accepts a price. The class is abstract but you can and should implement the method that returns the price information.

c. Implement the WalkupTicket class and the ComplementaryTicket class.

d. Implement the AdvanceTicket class. Provide a constructor that takes a parameter indicating the number of days in advance that the ticket is being purchased. Recall that the number of days of advanced purchase affects the ticket price.

e. Implement the StudentAdvanceTicket class. Provide a constructor that takes a parameter indicating the number of days in advance that the ticket is being purchased. The toString method should include a notation that this is a student ticket. This ticket costs half of an Advanceticket. If the pricing scheme for AdvanceTicket changes, the StudentAdvanceTicket price should be computed correctly with no code modification to the StudentAdvanceTicket class.

f. Write a main class TicketOrder that creates an array of tickets and show how polymorphism is working.