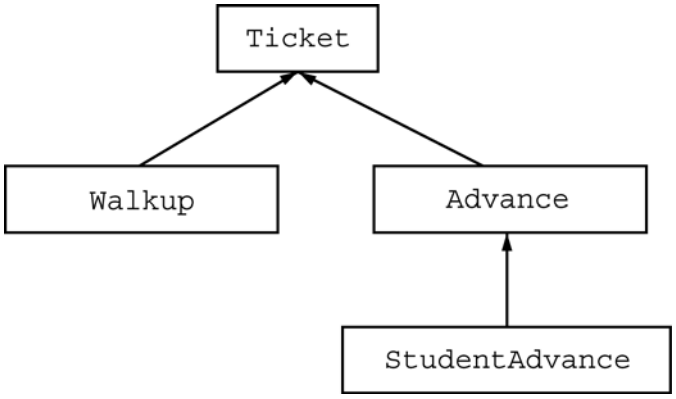
2. A set of classes is used to handle the different ticket types for a theater. The class hierarchy is shown in the following diagram.



All tickets have a serial number and a price. The class `Ticket` is specified as an `abstract` class as shown in the following declaration.

```
public abstract class Ticket
{
  private int serialNumber;   // unique ticket id number


  public Ticket()
  {  serialNumber = getNextSerialNumber();  }


  // returns the price for this ticket
  public abstract double getPrice();


  // returns a string with information about the ticket
  public String toString()
  {
    return "Number: " + serialNumber + "\nPrice: " + getPrice();
  }


  // returns a new unique serial number
  private static int getNextSerialNumber()
  {  /* implementation not shown */  }
}
```

Each ticket has a unique serial number that is assigned when the ticket is constructed. For all ticket classes, the `toString` method returns a string containing the information for that ticket. Three additional classes are used to represent the different types of tickets and are described in the table below.

Class	Description	Sample `toString` Output
`Walkup`	These tickets are purchased on the day of the event and cost 50 dollars.	`Number: 712` `Price: 50`
`Advance`	Tickets purchased ten or more days in advance cost 30 dollars. Tickets purchased fewer than ten days in advance cost 40 dollars.	`Number: 357` `Price: 40`

`StudentAdvance`	These tickets are a type of `Advance` ticket that costs half of what that `Advance` ticket would normally cost.	`Number: 134`
		`Price: 15`
		`(student ID required)`

Using the class hierarchy and specifications given above, you will write complete class declarations for the `Advance` and `StudentAdvance` classes.

(a) Write the complete class declaration for the class `Advance`. Include all necessary instance variables and implementations of its constructor and method(s). The constructor should take a parameter that indicates the number of days in advance that this ticket is being purchased. Tickets purchased ten or more days in advance cost $30; tickets purchased nine or fewer days in advance cost $40.

(b) Write the complete class declaration for the class `StudentAdvance`. Include all necessary instance variables and implementations of its constructor and method(s). The constructor should take a parameter that indicates the number of days in advance that this ticket is being purchased. The `toString` method should include a notation that a student ID is required for this ticket. A `StudentAdvance` ticket costs half of what that `Advance` ticket would normally cost. If the pricing scheme for `Advance` tickets changes, the `StudentAdvance` price should continue to be computed correctly with no code modifications to the `StudentAdvance` class.