

2005 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

1. In this question, you will implement two methods for a class `Hotel` that is part of a hotel reservation system. The `Hotel` class uses the `Reservation` class shown below. A `Reservation` is for the person and room number specified when the `Reservation` is constructed.

```
public class Reservation
{
    public Reservation(String guestName, int roomNumber)
    { /* implementation not shown */ }

    public int getRoomNumber()
    { /* implementation not shown */ }

    // private data and other methods not shown
}
```

An incomplete declaration for the `Hotel` class is shown below. Each hotel in the hotel reservation system has rooms numbered 0, 1, 2, . . . , up to the last room number in the hotel. For example, a hotel with 100 rooms would have rooms numbered 0, 1, 2, . . . , 99.

```
public class Hotel
{
    private Reservation[] rooms;
    // each element corresponds to a room in the hotel;
    // if rooms[index] is null, the room is empty;
    // otherwise, it contains a reference to the Reservation
    // for that room, such that
    // rooms[index].getRoomNumber() returns index

    private ArrayList waitList;
    // contains names of guests who have not yet been
    // assigned a room because all rooms are full

    // if there are any empty rooms (rooms with no reservation),
    // then create a reservation for an empty room for the
    // specified guest and return the new Reservation;
    // otherwise, add the guest to the end of waitList
    // and return null
    public Reservation requestRoom(String guestName)
    { /* to be implemented in part (a) */ }

    // release the room associated with parameter res, effectively
    // canceling the reservation;
    // if any names are stored in waitList, remove the first name
    // and create a Reservation for this person in the room
    // reserved by res; return that new Reservation;
    // if waitList is empty, mark the room specified by res as empty and
    // return null
    // precondition: res is a valid Reservation for some room
    //               in this hotel
    public Reservation cancelAndReassign(Reservation res)
    { /* to be implemented in part (b) */ }

    // constructors and other methods not shown
}
```

Copyright © 2005 by College Entrance Examination Board. All rights reserved.
Visit apcentral.collegeboard.com (for AP professionals) and www.collegeboard.com/apstudents (for AP students and parents).

GO ON TO THE NEXT PAGE.

2005 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

- (a) Write the Hotel method `requestRoom`. Method `requestRoom` attempts to reserve a room in the hotel for a given guest. If there are any empty rooms in the hotel, one of them will be assigned to the named guest and the newly created reservation is returned. If there are no empty rooms, the guest is added to the end of the waiting list and `null` is returned.

Complete method `requestRoom` below.

```
// if there are any empty rooms (rooms with no reservation),  
// then create a reservation for an empty room for the  
// specified guest and return the new Reservation;  
// otherwise, add the guest to the end of waitList  
// and return null  
public Reservation requestRoom(String guestName)
```

- (b) Write the Hotel method `cancelAndReassign`. Method `cancelAndReassign` releases a previous reservation. If the waiting list for the hotel contains any names, the vacated room is reassigned to the first person at the beginning of the list. That person is then removed from the waiting list and the newly created reservation is returned. If no one is waiting, the room is marked as empty and `null` is returned.

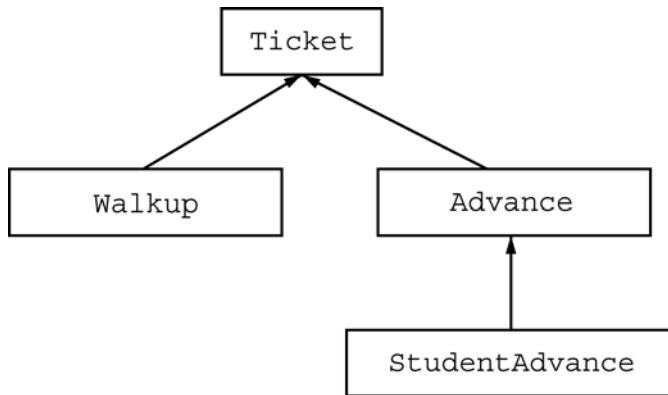
In writing `cancelAndReassign` you may call any accessible methods in the `Reservation` and `Hotel` classes. Assume that these methods work as specified.

Complete method `cancelAndReassign` below.

```
// release the room associated with parameter res, effectively  
// canceling the reservation;  
// if any names are stored in waitList, remove the first name  
// and create a Reservation for this person in the room  
// reserved by res; return that new Reservation;  
// if waitList is empty, mark the room specified by res as empty and  
// return null  
// precondition: res is a valid Reservation for some room  
//                 in this hotel  
public Reservation cancelAndReassign(Reservation res)
```

2005 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

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 */ }
}
```

**AP® COMPUTER SCIENCE A
2005 SCORING GUIDELINES**

2005 A Question 1: Hotel Reservation

Part A:	<code>requestRoom</code>	4 points
----------------	--------------------------	-----------------

- +1 loop over `rooms`
 - +1/2 attempt (must reference multiple elements of `rooms` in body)
 - +1/2 correct

- +1/2 test correct array entry for null (in context of loop)

- +1 1/2 handle new reservation (in context of a loop)
 - +1/2 attempt to create new reservation (some sense of `Reservation` construction)
 - +1/2 correctly create reservation (if add to `rooms`, must be in null location & assignment correct)
 - +1/2 return reservation (only if null entry)

- +1 handle wait list after loop or at appropriate time (only if full)
 - +1/2 add new guest to end of `waitlist` only once
 - +1/2 return null

Part B:	<code>cancelAndReassign</code>	5 points
----------------	--------------------------------	-----------------

- +1 look up room number
 - +1/2 attempt (must call `res.getRoomNumber()` or use loop to find `res`)
 - +1/2 correct (must call `res.getRoomNumber()`)

- +1/2 test `waitlist` to see if empty

- +2 1/2 handle nonempty `waitList`
 - +1/2 get *first* entry from `waitList` (only if `waitlist` is not empty)
 - +1/2 create new `Reservation` } *can get these points by*
 - +1/2 assign `Reservation` to correct room } *correctly calling requestRoom*
 - +1/2 remove only first entry from `waitlist` (only if `waitlist` is not empty)
 - +1/2 return new `Reservation` (only if `waitlist` is not empty)

- +1 handle empty case
 - +1/2 assign null to room (only if `waitList` is empty)
 - +1/2 return null (only if `waitList` is empty)

Note: If access using `get` on `rooms` is done more than once, deduct 1/2 usage point, not correctness (ditto for `set` on `rooms`).

Workshop Exam Materials
Canonical Solutions
2005 AP® Computer Science A

Question 1

PART A:

```
public Reservation requestRoom(String guestName)
{
    for (int i = 0; i < rooms.length; i++)
    {
        if (rooms[i] == null)
        {
            rooms[i] = new Reservation(guestName, i);
            return rooms[i];
        }
    }
    waitList.add(guestName);
    return null;
}
```

PART B:

```
public Reservation cancelAndReassign(Reservation res)
{
    int roomNum = res.getRoomNumber();
    if (waitList.isEmpty())
    {
        rooms[roomNum] = null;
    }
    else
    {
        rooms[roomNum] = new Reservation((String)waitList.get(0), roomNum);
        waitlist.remove(0);
    }
    return rooms[roomNum];
}
```

alternate solution

```
public Reservation cancelAndReassign(Reservation res)
{
    int roomNum = res.getRoomNumber();
    rooms[roomNum] = null;
    if (!waitList.isEmpty())
    {
        requestRoom((String)waitlist.get(0));
        waitlist.remove(0);
    }
    return rooms[roomNum];
}
```