SCHOOL OF COMPUTING & INFORMATION TECHNOLOGY

Programming 1 – Individual Project (30%) – AY 2017/18 Sem 2

HOTEL RESERVATION SYSTEM

SouthCoast Serenity Inn (SSI), a small hotel located on the southern point of the island, wishes to automate its operations which include reservations and booking of guests. The hotel has fifty (50) rooms, with various amenities included to cater to their various guests. SSI stores the following information on each room – room number (from 1 to 50), room type (smoking or non-smoking), a flat charge per room type ($4000 per day for a room which allows smoking and $3000 for non-smoking rooms) and a status which shows whether the room is available or has been reserved. Due to an overwhelming bumper tourist season, the maximum length that guests can stay at the inn is 7 days.

The hotel manager is seeking your assistance to develop an information system that can manage the reservations function. The development is to be done in two phases with specific requirements for each phase. Part 1 of the project required the use of loops to process a number of reservations made for guests. Part 2 requires the use of arrays.

Part 2

Given: March 21, 2018
Due: April 6, 2018 (submit to tutor on or before the Friday of this week)
Weighting: 15%

In this phase, the system requires a password in order to access the system. The program should be written such that the user has up to three attempts to log into the system. After three unsuccessful attempts, the user should be locked out.

Upon successful entry, provide an attractive menu to the user with basic menu options such as the one below:

SSI RESERVATION SYSTEM

- Add a Reservation
- Update a Reservation
- Display Total Estimated Revenue
- Exit

The requirements for each menu option are described below. Use attractive layouts for the display of information. Please note that the menu should be re-displayed on the screen after each menu selection until the user selects the Exit option.

Add a Reservation

Facilitate the entry of up to 10 reservations. Note that they do not all have to be entered at one time – for example, the user could enter 3 reservations sequentially then exit this option, display the estimated bills and re-select this option to **continue** entering the **remaining** reservations up to 10. Accept from the user the following data ONLY: *credit card number (use integer)*, *length of stay (in days)*, *room type*. Validate input for the length of stay (maximum 7 days) and room type (smoking or non-smoking) according to the previous requirements. Use parallel arrays to store the data. Determine for each guest the expected bill which includes the cost for the number of days stayed along with a 15% room tax. Produce an attractive output showing the room number, room type (this corresponds to the position entered in the array – the first record stored in the array would automatically be assigned room number 1), length of stay, cost for the days, room tax, and total cost of stay. *(Note that the cost for the days, room tax, and total cost of stay are not to be stored in the array, merely calculate and display these values).*

Update a Reservation

Prompt the user for the credit card number. If found, the program should display the details of the reservation – length of stay and room type. Only the length of stay and room type can be updated. Allow the user to update either one or both of those fields and display the modified reservation. Also display the expected bill and output as in the previous option. If the credit card number has not been found in the array, display an appropriate error message.

Display Total Estimated Revenue

For all the reservations entered in the array, output the following: total number of reservations, total charges for the cost of stay of all guests, room tax amounts expected for all guests and the total estimated bill for all guests. Also produce the total estimated bill for each room type and the average length of stay based on all reservations.

Your program should assume that credit card numbers will be unique – that is, the same credit card number will not be used to make another reservation.

Required:

i. The pseudocode which correctly expresses the logic for solving the problem as expressed above.
ii. The C code providing a correct solution and corresponding to the pseudocode produced

Mark Scheme

Deliverable	**Marks**	**Notes**
1. Pseudocode (accuracy of logic)	20	
2. C program (fully documented – purpose, meaningful comments throughout – especially for code which is not naturally understood; meaningful variable names etc., efficient code i.e. little or no redundancy of code, appropriate structures, constants etc.)	20	
3. Attractiveness of User Interface	5	
4. Overall Presentation (timely delivery, neatness, readability)	3	
5. INTERVIEW/PRESENTATION OF PROJECT (The full marks for this section will be deducted for persons who miss the interview)	**10**	
6. Declaration of Authorship Submission	2	
Total	**60**	

NOTE: Please review the University's policy on plagiarism.
Copying/reproducing/submitting another person's assignment/test is a MAJOR offence according to the University's regulations governing Academic Misconduct. Any evidence of work suspected to be copied from another student will be either be assigned a 0 or if egregious, will be treated according to the procedural rules of the Academic Misconduct Policy. Please note that merely changing the variable names from another program is still an offence. Your instructors have vast experience in determining the similarity of code.
If you do not understand or cannot complete the program accurately, it is better to submit the parts that you understand and be graded accordingly, than to "try to fool" the instructor and submit work that is not your own. Work that may have been done by an outside "expert" that cannot be reasonably explained is ALSO subject to scrutiny.