**National University of Computer and Emerging Sciences**



|  |  |  |
| --- | --- | --- |
|  | **OOAD** | |
|  | **Lab Manual 8** | |
|  |  |  |
| Course Instructor |  | Sir Amir Raheem |
|  |  |  |
| Lab Instructor(s) |  | Asad Ullah |
|  |  | Farwa Batool |
|  |  |  |
| Section |  | B |
|  |  |  |
| Date |  | 11/12/2018 |
|  |  |  |
| Semester |  | Fall 2018 |
|  |  |  |

Department of Computer Science

FAST-NU, Lahore, Pakistan

Keep the following good programming practices in mind when writing your code:

* Comment your code intelligently.
* Indent your code properly.
* Use meaningful variable names.
* Use meaningful prompt lines/labels for input/output.
* use meaningful project and JAVA file name
* create separate java classes for each task

**(Swing and Event Driven Programming)**

The uml diagram in this question is to introduce some common terms, "dictionary" for online

shopping - Customer, Web User, Account, Shopping Cart, Product, Order, Payment, etc. and

relationships between. It could be used as a common ground between business analysts and

software developers.

Each customer has unique id and is linked to exactly one account. Account owns shopping cart

and orders. Customer could register as a web user to be able to buy items online. Customer is not

required to be a web user because purchases could also be made by phone or by ordering from

catalogues. Web user has login name which also serves as unique id. Web user could be in

several states - new, active, temporary blocked, or banned, and be linked to a shopping cart.

Shopping cart belongs to account.

Account owns customer orders. Customer may have no orders. Customer orders are sorted and

unique. Each order could refer to several payments, possibly none. Every payment has unique id

and is related to exactly one account

Consider a reservation system for an inter-city transportation company that operates buses on

different routes. Each route is determined by a source and destination. On each route different

buses run at different timings. There are two categories of buses that differ in services and

consequently fare. Economy buses provide standard transportation facility and their fare is

computed as a product of the route distance and base rate (fare per km). Luxury buses on the

other hand have a higher base rate, considering reduced seating capacity. In addition, luxury

buses provide different options for refreshments as well as extra luggage, the cost of which can

be added to the fare.

Consider a reservation system for an inter-city transportation company that operates buses on

different routes. Each route is determined by a source and destination. On each route different

buses run at different timings. There are two categories of buses that differ in services and

consequently fare. Economy buses provide standard transportation facility and their fare is

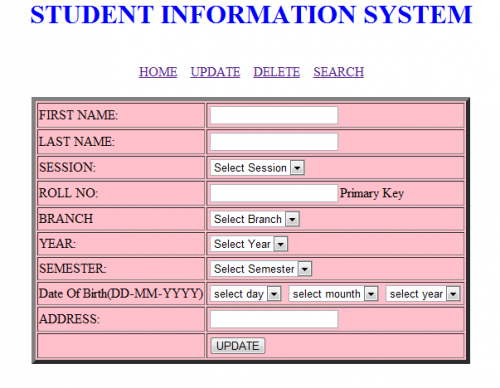
computed as a product of the route distance and base rate (fare per km). Luxury buses on the

other hand have a higher base rate, considering reduced seating capacity. In addition, luxury

buses provide different options for refreshments as well as extra luggage, the cost of which can

be added to the fare.

**Task 1:** Create the following GUI for a student information system using java swing components.



**Task 2:**

Create a class Student having following data members.

* Firstname
* LastName
* Session
* RollNumber
* Branch
* Year
* Semester (integer between 1 to 8)
* Date of birth
* Address

Now create an class with name as Record Student. This class has the following data members.

* List<Student > students
* int count\_of\_Students

**Functions**

1. Boolean Add\_Student (Student m)

* This function will take student object and add them to the list.
* Returns true in case of success.

1. Boolean Delete\_Student (String Rollnumber)

* This function will take student rollnumber and search for it in the list and delete the record in case of record found and return true.

1. Boolean Search (String Rollnumber)

* This function will take student rollnumber and search for it in the list and if found returns true.

**Task 3: (Event Handling)**

For the GUI you have built in task 1 you have to implement event listeners for buttons update, delete and Search using task2 functions you have implemented.

-------------------------------------------------------------Good Luck----------------------------------------------------------------