United International University (UIU)
Dept. of Computer Science and Engineering (CSE)
CSE 1116, Object Oriented Programming Lab
Spring 2024, MoI

Practice Sheet - 2

Topics : Abstract classes, Interfaces, File I/O, Exception, GUI, ArrayList and Sorting

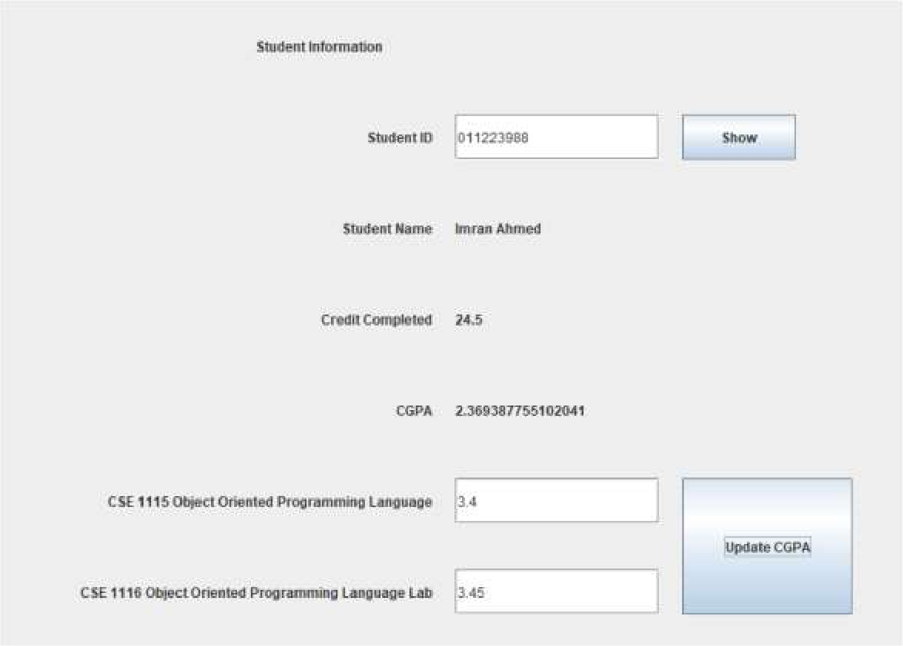
Problem1:

Create a student information portal like below. We can complete two tasks.

Task 1: If we insert student id, then press the show button, then student name, credit and cgpa will get updated from a file named “students.txt” containing student information. File example is given below.

```
011181202,Mahbub Hossain,15,3.67
011223988,Imran Ahmed,20,2.56
```

Task 2: If we insert students' expected gpa of OOP Theory and Lab courses, then the cgpa and credit completed will get updated correspondingly to the students information window.



Student ID 011223988

Student Name Imran Ahmed

Credit Completed 24.5

CGPA 2.369387755102041

CSE 1115 Object Oriented Programming Language 3.4

CSE 1116 Object Oriented Programming Language Lab 3.45

Problem2:

Handling Wrong Name: Write a java code that takes as input, names of people. If the name starts with adigit (0-9)(ASCII value 48-57), the method will throw a user-defined exception named ‘**WrongNameException’**. Write necessary try-catch block and user-defined Exception class to handle the exception. If exception is found, it will replace the digit with a blank space(‘ ‘) and print the name. Otherwise, it will show the name as output. You can use the replace method in your program.
Syntax: Variable_name.**replace**(char **searchChar**, char **newChar**);

Sample input:
James Bond
7George F. Kennedy
Sample output:
James Bond
George F. Kennedy

Problem3:

1. Create the Product.txt file. The file contains the information of a product of a stationary shop in every line.
2. The format for a product is: <Product_Name>, <Product_id>, <Price>,<Quantity>
    a. From your Main Class, **read every line of the file**
    b. Create a Product object with this information.
3. Store the object in an ArrayList that can hold Product objects. Along with previous content add another field <Product Brand> for Product Objects.
4. Now **write the all information in a Separate file** called **NewProduct.txt**.
5. Check if the list is empty; **Create a GUI, use a Search button to check if a specific Product is in the list** (Search by Product name).
6. **Show the product in a TextArea**.
7. Sort the ArrayList containing Product objects according to Price. Print out this sorted ArrayList.

Product.txt file-

```
Surf Excel, DTR102,40,2
Cucumber,VEG078,15,5
Eggs,GRC123,12,12
```

Problem4:

There is a magical world of Narnia, where time is different from the time in this world and where animals can speak. The path to Narnia is through a cupboard. A very special cupboard.
In the magical world of Harry Potter, there is a room called Room of Requirement. This is like a cupboard too- it has a lot of items stored in it.

Cupboard has the ability to open doors to places. Each time cupboard is opened visited place is increased by one unit. Cupboards can also open the door to the magical places. But each cupboard leads to different places. So there is no way to define the function travel(MagicalPlace) generally, for a cupboard. If we specify the cupboard then we can say how travel(MagicalPlace) is implemented. Cupboards also have a *visitedPlace* variable which keeps count of the number of magical places visited until now.

MagicalPlace is implemented from an interface called Place. Place has methods isMagical() and expectedDistance(). For any place, the expected distance is double of the number of letters in the place's name. isMagical() returns true if the place is magical, false otherwise. For any place, the variables ***name, placeType*** and ***password*** are set during initialization.

CupboardNarnia can only open doors if the password is "Narnia". Every time travel method is called, it prints "Welcome to Narnia" if the password is right. In this travel(MagicalPlace p) function, check if the password is right and then print the line and increase the ***visitedPlace*** variable by 1.

RoomOfRequirement can open doors to every place except those which are not magical which it determines by using the isMagical() method. Every time ***travel*** is called, it prints the name of the place it opened door to. In this travel(MagicalPlace p) function, check if a place is magical or not by calling isMagical(); if true, print the name of the place and increase the ***visitedPlace*** variable by 1.

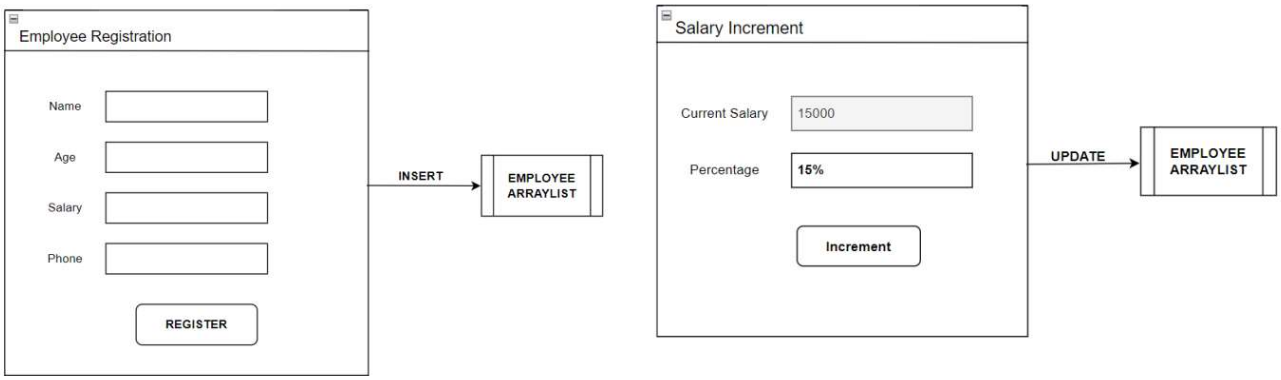
Main Class Code's been given below. Generate output for this code:

```
MagicalPlace place1 = new MagicalPlace(“Hogwarts”, “magical”,
“Ravenclaw”);

MagicalPlace place2 = new MagicalPlace (“Narnia”, “not magical”,
“Narnia”);

CupboardNarnia cn = new CupboardNarnia (10);
//visited place is 10 Initially
cn.travel(place1);
cn.travel(place2);
System.out.println(cn); //prints number of visited place
hereRoomOfRequirement r = new RoomOfRequirement (50);
// visited place is 50 initially.
r.travel(place1);
r.travel(place2);System.out.println(r); // prints number of visited
place here.
```

Problem5:



Now the functionalities for this Java Swing program will be the following:

a. First, it will open a frame for **Employee Registration** that will take their

  i. Name
  ii. Age
  iii. Salary
  iv. Phone number

b. After that, it will insert these data inside an ArrayList which will be declared inside the Main class.

c. After **REGISTER**, the Employee Registration will become invisible and open the **Salary Increment** frame.

d. Inside the Salary Increment frame, the current salary will be shown and it will be Non-editable.

e. Also, there will be a percentage to input. After pressing increment, the salary will be **increased** accordingly and get updated inside the ArrayList.