ID: 1930319

Name: Mohammad Arshad Hossain Rabul

CSE 213 Final exam, Autumn 2021, Section: 1

Project Title: Textile Factory

1) a)

User: Supplier

Goal: Check Orders

Clicking on this goal will show all orders placed by the company in a tabular form. The table will contain four columns ("Order no.", "Product ~~Name~~ type", "quantity" and "deadline"). In this case the inputs will be product name (selected from combo box).

b) Clicking on check order will show an UI where it will have a ~~check~~ combo box where the user will be able to choose ("All", "Machinery", "silk", "wool", "cotton", "Rayon", "Polyester", "Nylon"). After choosing any one option there will be a button "show", will generate the table by ~~matching~~ matching the chosen option ~~to~~ from the file "Orderlist.txt".

Example

Order No.	Product type	Quantity	Deadline
1	Wool	100 kg	25 Dec 2021

c) Factory manager will write the content of the file ("Orderlist.txt").

User : Factory manager
Goal : Order raw material

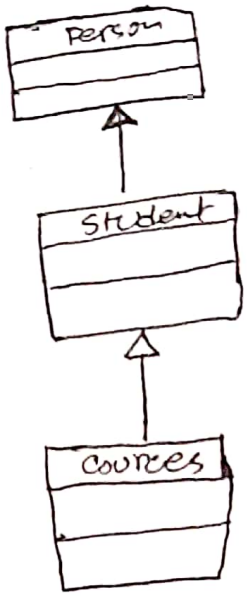
- Clicking on "order raw material" will take the user to an UI which shows the buttons named the seetors (Mechines, silk, wool, cotton, polyester, Nylon).
- Clicking on the listed button will take the user to another UI where it shows the materials listed by the users.
- Clicking on the desired material will show a dialogue box which has fields for amount of that raw material to order and on the top there will be quantity available 'lable' which will show the amount of materials available.
- After filling the field there will be a button confirm orde, clicking on it will ask for confirmation (as warning message) after clicking on "ok" the order will be appended on the file and will be available to the supplier.

```
@FXML
Private ComboBox combobox;
@FXML
Private TableView<orders> tableview;
@FXML
Private Tablecolumn<Order, string> firstcolumn;
@FXML Private Table column<orders, string> second column;
@FXML Private TableColumn<order, string> third column;
@FXML Private Table column <order, string> fourth column;

@Override
Public void initialize (URL.url, ResourceBundle rb)
{
    combobox.getItems().addAll ("Machine", "wool", "silk",
                "cotton", "Nylon", "Polyester", "Rayon");
}

@FXML
Private void showButton (MouseEvent event)
{
    ArrayList<order> orders;
    String line; String[] tokens;
    File f = new File ("orderlist.txt");
    Scanner s = new scanner(f);
    while (s.hasNextline())
    {
        Line = s.nextline();
        tokens = line.split(",");
```

```
if(combobox.getValue().equal (tokens[1])
{
        order.add ( tokens) ;
 }

for( order s : tokens)
{   first column. get


}
```

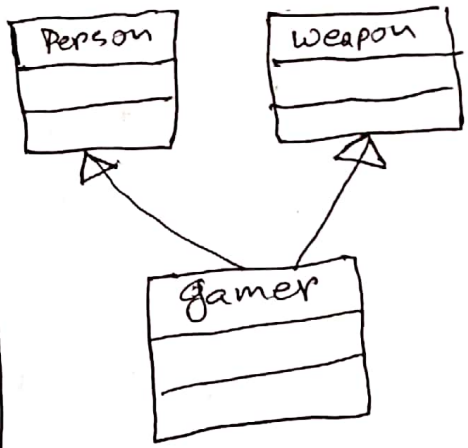
(3) a).

```
Class person
{

}

class student extends person
{

}

class cources extends student
{

}
```



multilevel inheritance

```
Class Person
{

}

class weapon
{

}

class gamer extends person, weapon
{

}
```



multiple inheritance

When one class inherits another class and the inherited child class is inherited by another class we call it multi-level inheritance and when one class inherits two or more class we call it multiple inheritance.

b) we use non-abstract method in an abstract class to pre-define the method ~~and~~, which can be used without overriding if ~~it is~~ the class is inherited

~~Abstract~~

```
abstract class person
{  + name, - Id;
    Void displayInfo()
    {
        sout("Name:"+ name + "ID;"+ Id);
    }

    void calculate();
}

~~Student~~ class student extend person
        {
            display Info();
        }
```

9) Marker interface is an empty interface.
They are use for seriable, cloneable, etc.