

KSU/CCIS/CS	CSC 113 - Final	Fall 30/31
Name:	Student ID:	Dr:

### Question1 :

Find out the syntax errors of the following program :

```
//=====file Test2Exception.java =====
public class Test2Exception {
    private int val;
    public Test2Exception(int n){
        val = -n;
    }
    public int IsThereProblem() throws ArithmeticException{
        try {
            -val = val;          // expression
            if ((val > 0){
                System.out.println("yes There is a Problem "+Math.sqrt(val) ) ;
                return 4.5* val ;          // int return double
            }
            System.out.println("No Problem Everything is OK");
            throw new ArithmeticException() // --- ;
        }
        catch ( ArithmeticException exception ) {
            System.out.println("Problem before the log operation"); // string no ""
            throw new ArithmeticException();
        }
    }
    public void catcher() thws ArithmeticException {      // throws
        try {
            double x =IsThereProblem();
            System.out.println("The result is : "+ val+" ==== "+x);
        }
        catch ( ArithmeticException e ){
            System.out.println("Exception handled in the catcher" );
            throw new ArithmeticException();
        }
        return val ;          // void return int
    }
}

//=====file ExceptionTestMain.java =====
public class ExceptionTestMain {
    public static void main(Sting[] args) {          // String
        Test2Exception a ;
        try {
            a.catcher();          // no initialization
        } catch (ArithmeticException e) {
            System.out.print("Problem with the catcher");
        }
    }
}
```

KSU/CCIS/CS	CSC 113 - Final	Fall 30/31
Name:	Student ID:	Dr:

### Question1 :

Find 8 syntax errors in the following program :

```

public class Test2Exception {
    private int val;
    public Test2Exception(double n){
        val = -n; // double to int
    }
    public int IsThereProblem() throws ArithmeticException{ // throws
        try {
            val = -val;
            if (val > 0){
                System.out.println("yes There is a Problem "+Math.sqrt(val) ) // ;
                return 2.5* val ; // int return double
            }
            System.out.println("No Problem Everything is OK"); // string no ""
            throw new ArithmeticException();
        }
        catch ( ArithmeticException exception ) {
            System.out.println( "Problem before the log operation" );
            throw new ArithmeticException();
        }
    }
    public void catcher() throws ArithmeticException {
        try {
            double x =IsThereProblem();
            System.out.println("The result is : "+ val+" ==== "+x);
        }
        catch ( ArithmeticException e ){
            System.out.println("Exception handled in the catcher" );
            throw new ArithmeticException();
        }
        return val ; // void return int
    }
}

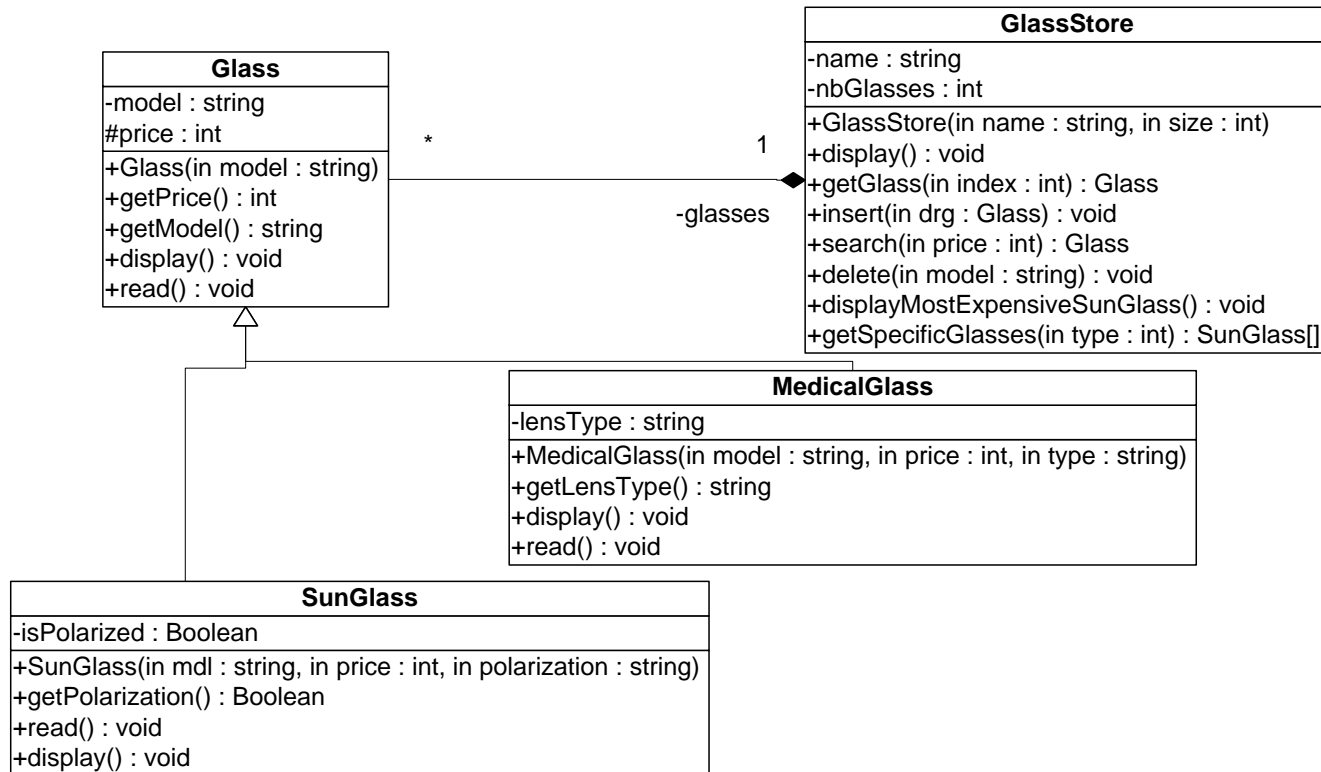
//===== Main =====
public class ExceptionTestMain { // typos main
    public static void man(String[] args) {
        Test2Exception a ;
        try {
            a.catcher(); // no initialization
        } catch (ArithmeticException e) {
            System.out.print("Problem with the catcher");
        }
    }
}

```

<b>KSU/CCIS/CS</b>	<b>CSC 113 - Final</b>	<b>Fall 30/31</b>
<b>Name:</b>	<b>Student ID:</b>	<b>Dr:</b>

## Question 2 :

Given the following UML diagram:



Where :

*getGlass(int)*: it returns the glass located in the place *index*

*insert(Glass)*: it adds a new glass. As it is described in the diagram the relationship is a **composition**.

*search(int)*: it returns the first glass verifying the required parameter.

*delete(Glass)*: it deletes the first glass verifying the required parameter.

*displayMostExpensiveSunGlass()*: it displays the data of the sun glass that is the most expensive.

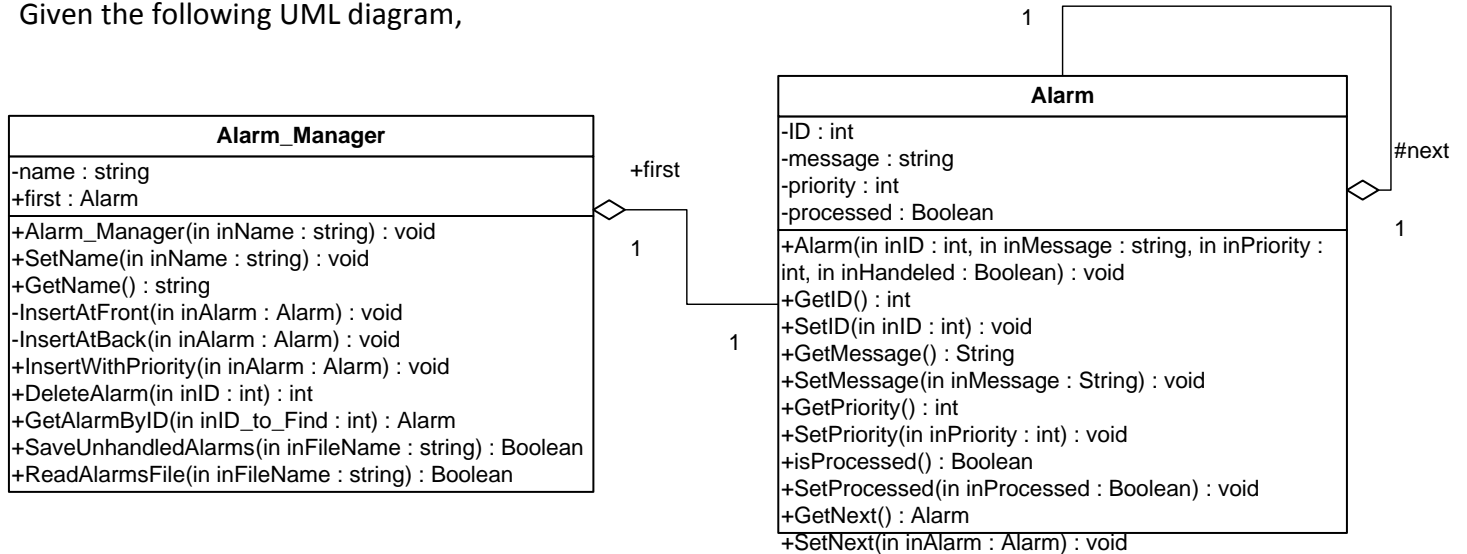
*getSpecificGlasses(int)*: (1) it creates an array of glasses. (2) it stores in it all the glasses verifying the required parameter; then it returns this created array .

- Write in Java the *Glass*, the *MedicalGlass* and the *GlassStore* classes

<b>KSU/CCIS/CS</b>	<b>CSC 113 - Final</b>	<b>Fall 30/31</b>
<b>Name:</b>	<b>Student ID:</b>	<b>Dr:</b>

### Question3 :

Given the following UML diagram,



**Implement ONLY the class Alarm\_Manager, Knowing that:**

=> In class Alarm:

- *priority*: can have value 1, 2 or 3.

=> In the class Alarm\_Manager, the different methods should be implemented as follows:

- *InsertAtFront()* : Inserts the given Alarm object at the front of the linked list
- *InsertAtBack()* : Inserts the given Alarm object at the end of the linked list.
- *InsertWithPriority()* : the given Alarm is inserted in the linked list as following:
  - If Alarm's priority == 3, the Alarm is inserted at the front of the linked list.
  - If Alarm's priority == 1, the Alarm is inserted at the back of the linked list.
  - If Alarm's priority == 2, the Alarm is inserted after Alarms of priority 3 and before Alarms of priority 1.
- *DeleteAlarm()* : removes from the linked list the Alarm that has the given ID.
- *GetAlarmByID()* : returns the Alarm that has the given ID or NULL if not found in Alarm\_Manager.
- *SaveUnhandledAlarms()* : Saves to a file all the Alarms of the list that are not handled yet.
- *ReadAlarmsFile()* : reads all the alarms in the file and inserts in the Alarm\_Manager ONLY the alarms that have *priority* == 3