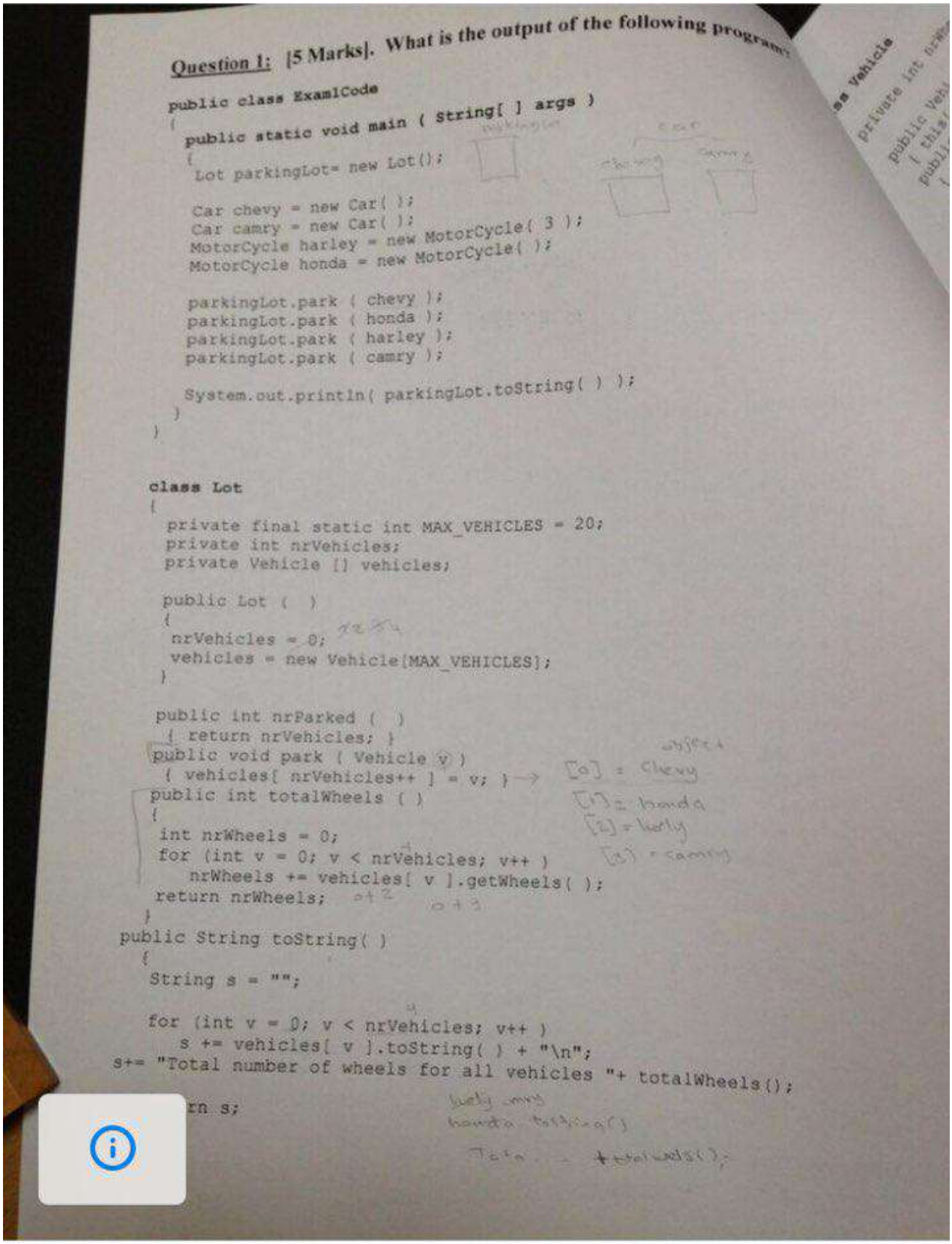
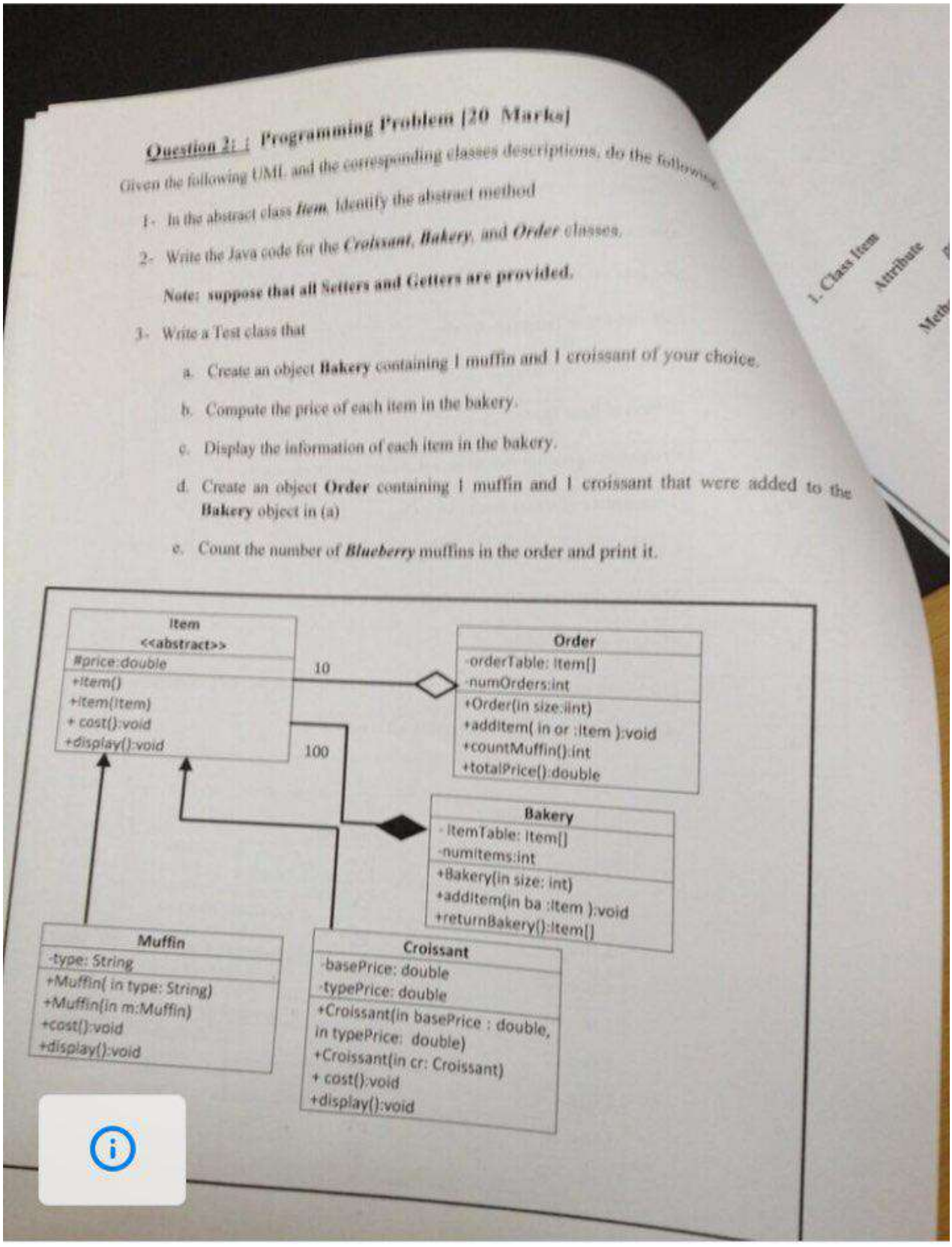


Question 1: [5 Marks]. What is the output of the following program?

```
public class ExamCode
{
  public static void main ( String[ ] args )
  {
   Lot parkingLot= new Lot();

   Car chevy = new Car( );
   Car camry = new Car( );
   MotorCycle harley = new MotorCycle( 3 );
   MotorCycle honda = new MotorCycle( );

   parkingLot.park ( chevy );
   parkingLot.park ( honda );
   parkingLot.park ( harley );
   parkingLot.park ( camry );

   System.out.println( parkingLot.toString( ) );
  }
}

class Lot
{
  private final static int MAX_VEHICLES = 20;
  private int nrVehicles;
  private Vehicle [] vehicles;

  public Lot ( )
  {
   nrVehicles = 0;
   vehicles = new Vehicle[MAX_VEHICLES];
  }

  public int nrParked ( )
   { return nrVehicles; }
  public void park ( Vehicle v )
   { vehicles[ nrVehicles++ ] = v; }
  public int totalWheels ( )
  {
   int nrWheels = 0;
   for (int v = 0; v < nrVehicles; v++ )
      nrWheels += vehicles[ v ].getWheels( );
   return nrWheels;
  }
public String toString( )
  {
   String s = "";

   for (int v = 0; v < nrVehicles; v++ )
      s += vehicles[ v ].toString( ) + "\n";
s+= "Total number of wheels for all vehicles "+ totalWheels();
   rn s;
```

0 comments Activity

Type a comment



Question 2: : Programming Problem [20 Marks]

Given the following UML and the corresponding classes descriptions, do the following:

1- In the abstract class ***Item***, Identify the abstract method

2- Write the Java code for the ***Croissant***, ***Bakery***, and ***Order*** classes.

Note: suppose that all Setters and Getters are provided.

3- Write a Test class that

a. Create an object **Bakery** containing 1 muffin and 1 croissant of your choice.

b. Compute the price of each item in the bakery.

c. Display the information of each item in the bakery.

d. Create an object **Order** containing 1 muffin and 1 croissant that were added to the **Bakery** object in (a)

e. Count the number of ***Blueberry*** muffins in the order and print it.

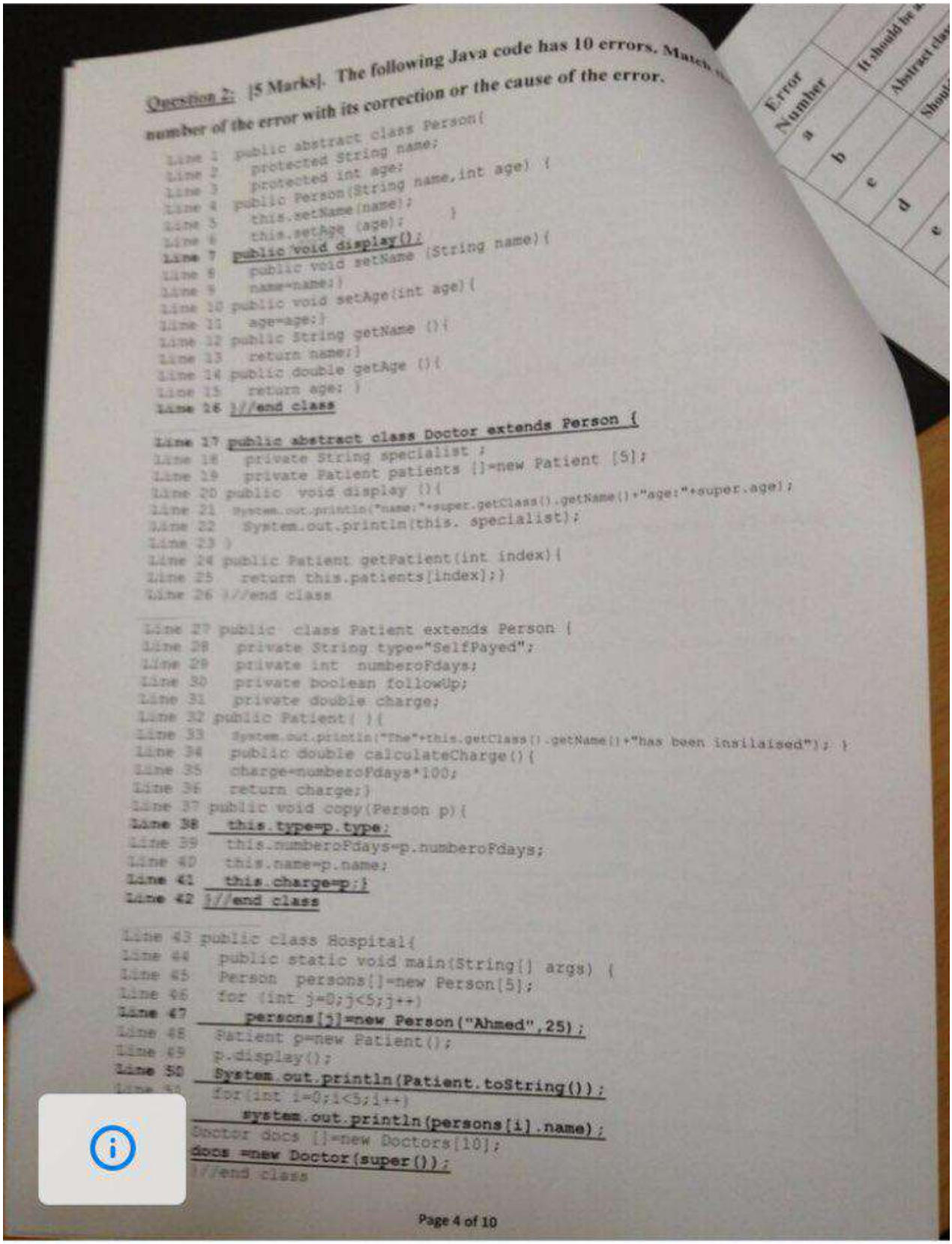
1. Class Item

Attribute

Meth

٪ ٥٤ ١٢:٥١ ص 4G تيليجرام

Mid 1 3 of 9 •••



Question 2: [5 Marks]. The following Java code has 10 errors. Match ... number of the error with its correction or the cause of the error.

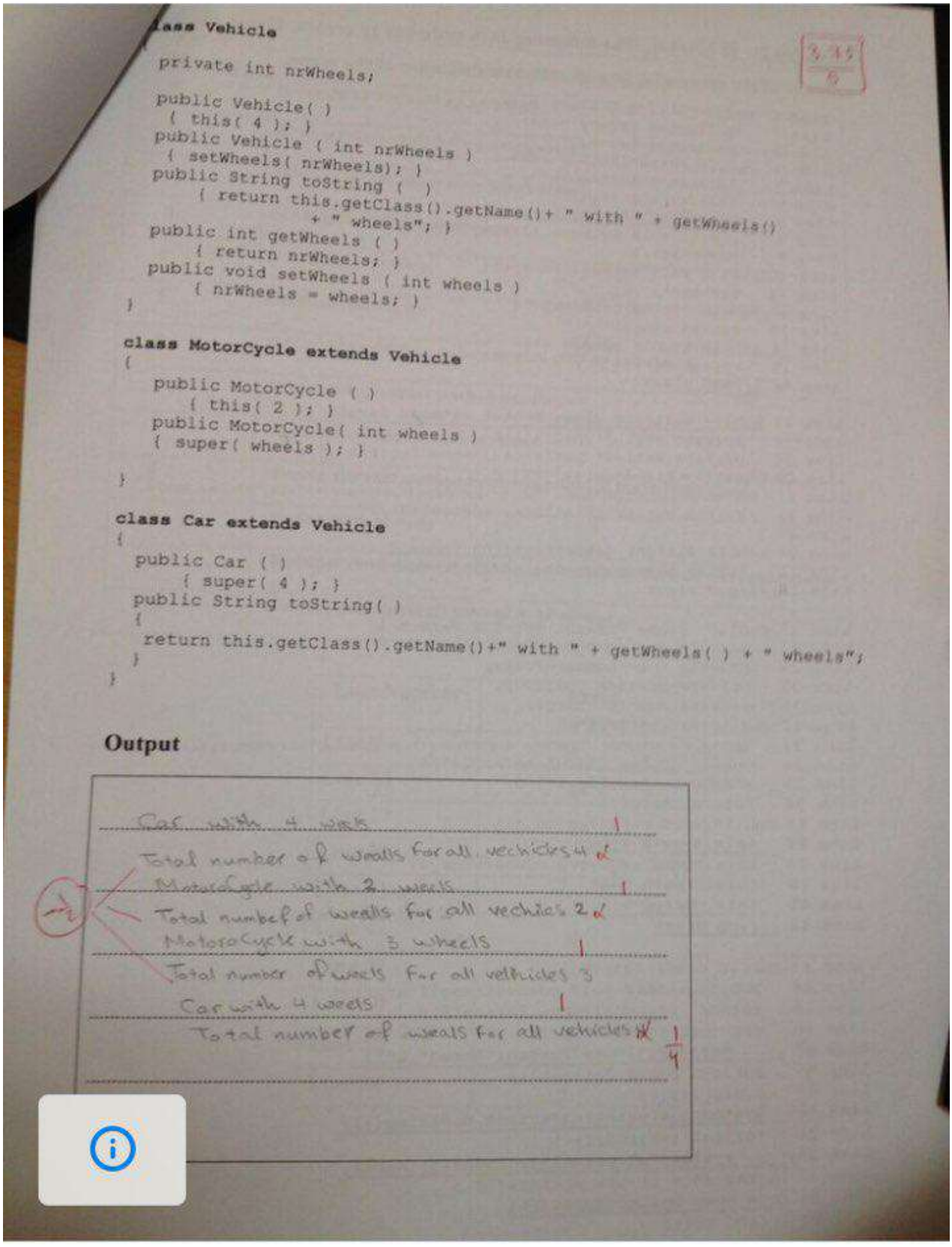
Error Number	
a	
b	
c	
d	
e	

```
Line 1  public abstract class Person{
Line 2    protected String name;
Line 3    protected int age;
Line 4  public Person(String name,int age) {
Line 5    this.setName(name);
Line 6    this.setAge (age);    }
Line 7  public void display();
Line 8    public void setName (String name){
Line 9    name=name;}
Line 10 public void setAge(int age){
Line 11   age=age;}
Line 12 public String getName (){
Line 13   return name;}
Line 14 public double getAge (){
Line 15   return age; }
Line 16 }//end class

Line 17 public abstract class Doctor extends Person {
Line 18   private String specialist ;
Line 19   private Patient patients []=new Patient [5];
Line 20 public  void display (){
Line 21   System.out.println("name:"+super.getClass().getName()+"age:"+super.age);
Line 22   System.out.println(this. specialist);
Line 23 }
Line 24 public Patient getPatient(int index){
Line 25   return this.patients[index];}
Line 26 }//end class

Line 27 public  class Patient extends Person {
Line 28   private String type="SelfPayed";
Line 29   private int  numberofdays;
Line 30   private boolean followUp;
Line 31   private double charge;
Line 32 public Patient( ){
Line 33   System.out.println("The"+this.getClass().getName()+"has been insilaised"); }
Line 34   public double calculateCharge(){
Line 35   charge=numberofdays*100;
Line 36   return charge;}
Line 37 public void copy(Person p){
Line 38   this.type=p.type;
Line 39   this.numberofdays=p.numberofdays;
Line 40   this.name=p.name;
Line 41   this.charge=p;}
Line 42 }//end class

Line 43 public class Hospital{
Line 44   public static void main(String[] args) {
Line 45   Person  persons[]=new Person[5];
Line 46   for (int j=0;j<5;j++)
Line 47     persons[j]=new Person("Ahmed",25);
Line 48   Patient p=new Patient();
Line 49   p.display();
Line 50   System.out.println(Patient.toString());
         for(int i=0;i<5;i++)
           system.out.println(persons[i].name);
Doctor docs []=new Doctors[10];
docs =new Doctor(super());
}//end class
```

Page 4 of 10

0 comments Activity

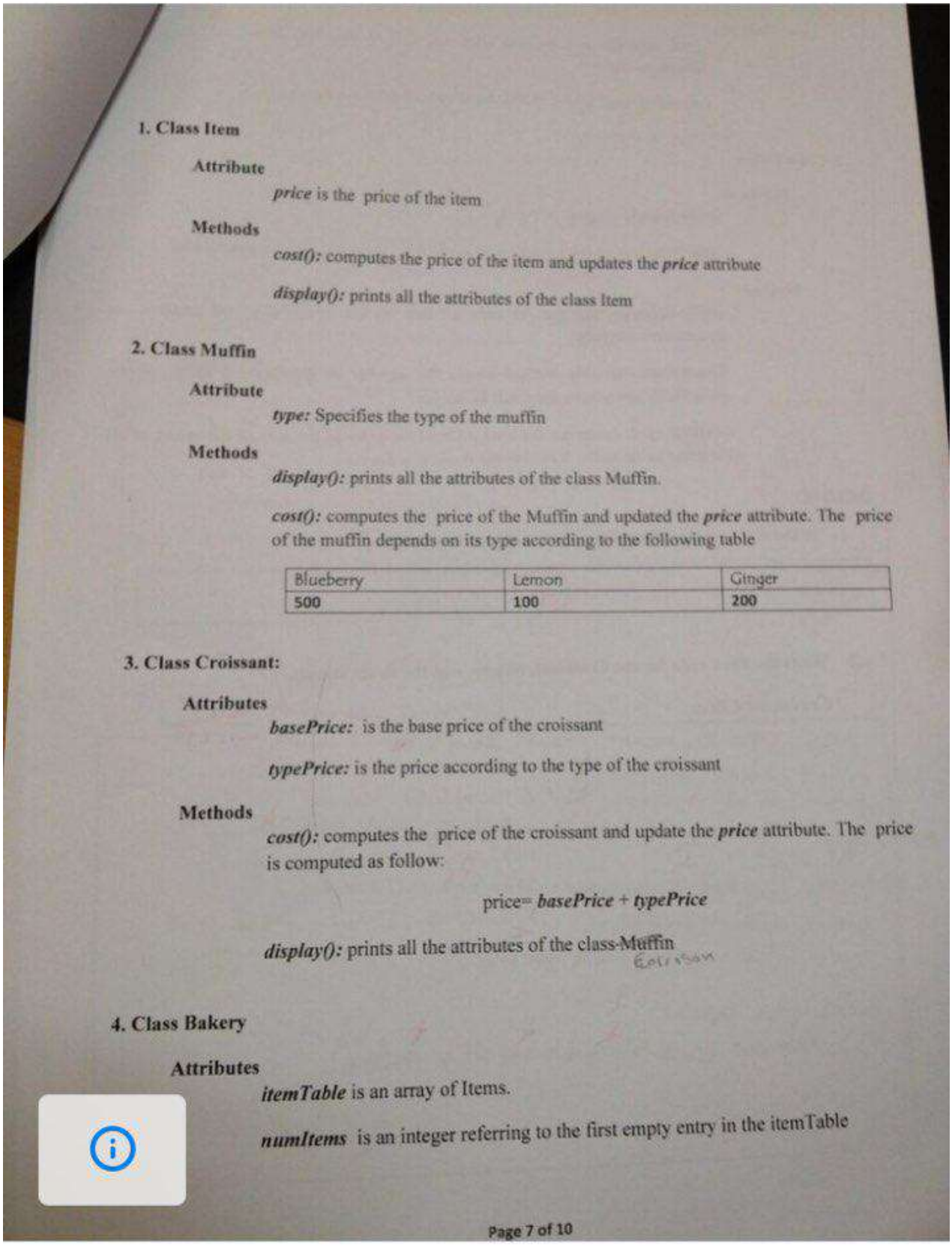
Type a comment



```
class Vehicle
{
   private int nrWheels;

   public Vehicle( )
     { this( 4 ); }
   public Vehicle ( int nrWheels )
     { setWheels( nrWheels); }
   public String toString ( )
       { return this.getClass().getName()+ " with " + getWheels()
                + " wheels"; }
   public int getWheels ( )
       { return nrWheels; }
   public void setWheels ( int wheels )
       { nrWheels = wheels; }
}

class MotorCycle extends Vehicle
{
   public MotorCycle ( )
      { this( 2 ); }
   public MotorCycle( int wheels )
   { super( wheels ); }

}

class Car extends Vehicle
{
  public Car ( )
      { super( 4 ); }
  public String toString( )
  {
   return this.getClass().getName()+" with " + getWheels( ) + " wheels";
  }
}
```

Output

Car with 4 wheels 1
Total number of wheels for all vechicles 4 ✓
Motorcycle with 2 wheels 1
Total number of wheels for all vechicles 2 ✓
Motorcycle with 3 wheels 1
Total number of wheels for all vehicles 3
Car with 4 wheels 1
Total number of wheels for all vehicles ✗ 1/4



1. Class Item

Attribute

price is the price of the item

Methods

cost(): computes the price of the item and updates the ***price*** attribute

display(): prints all the attributes of the class Item

2. Class Muffin

Attribute

type: Specifies the type of the muffin

Methods

display(): prints all the attributes of the class Muffin.

cost(): computes the price of the Muffin and updated the ***price*** attribute. The price of the muffin depends on its type according to the following table

Blueberry	Lemon	Ginger
500	100	200

3. Class Croissant:

Attributes

basePrice: is the base price of the croissant

typePrice: is the price according to the type of the croissant

Methods

cost(): computes the price of the croissant and update the ***price*** attribute. The price is computed as follow:

price= ***basePrice*** + ***typePrice***

display(): prints all the attributes of the class Muffin

4. Class Bakery

Attributes

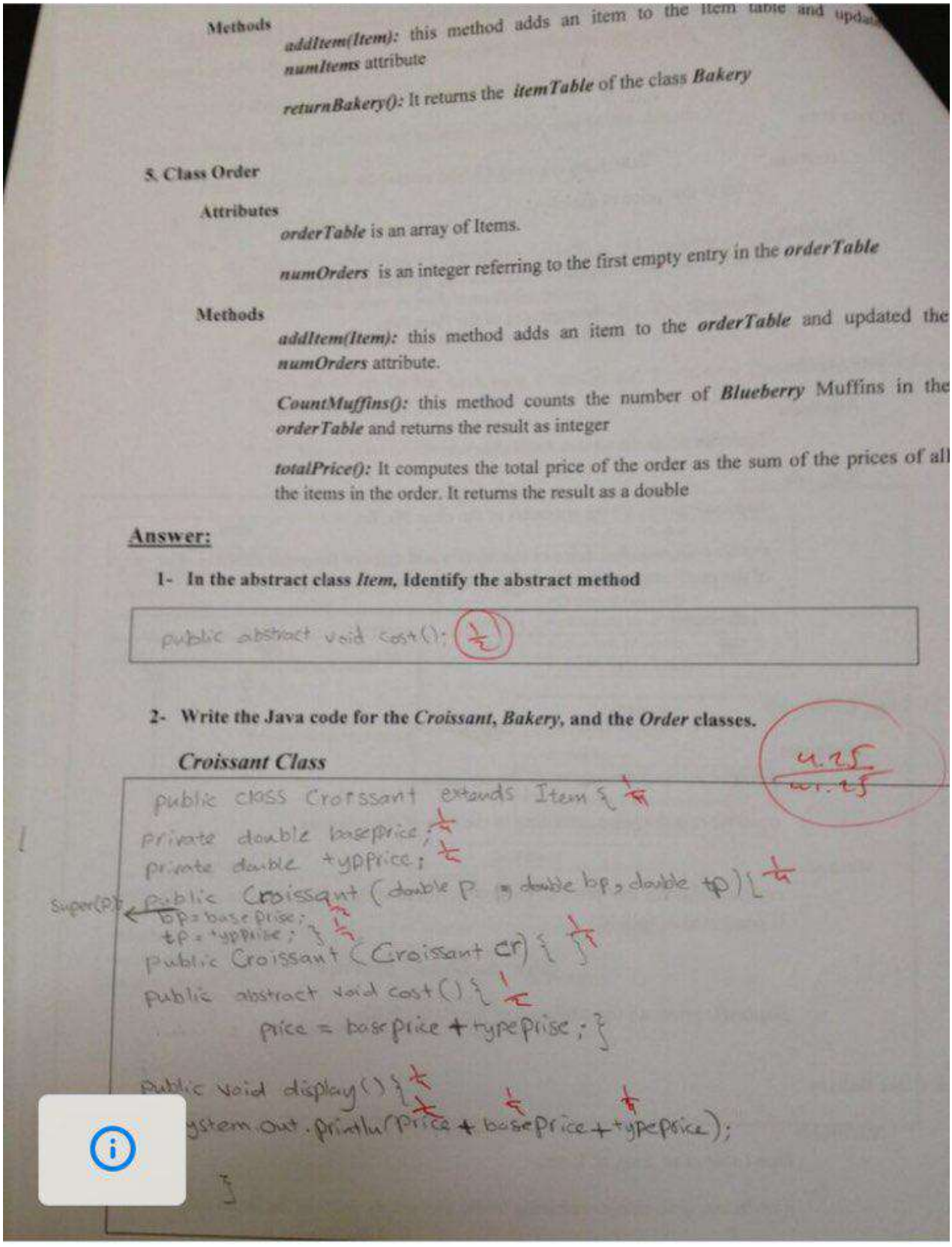
itemTable is an array of Items.

numItems is an integer referring to the first empty entry in the itemTable

Page 7 of 10

٪ ٥٥ ١٢:٥١ ص 4G تيليجرام

Mid 1 7 of 9 •••



Methods

addItem(Item): this method adds an item to the item table and updates the ***numItems*** attribute

returnBakery(): It returns the ***itemTable*** of the class ***Bakery***

5. Class Order

Attributes

orderTable is an array of Items.

numOrders is an integer referring to the first empty entry in the ***orderTable***

Methods

addItem(Item): this method adds an item to the ***orderTable*** and updated the ***numOrders*** attribute.

CountMuffins(): this method counts the number of ***Blueberry*** Muffins in the ***orderTable*** and returns the result as integer

totalPrice(): It computes the total price of the order as the sum of the prices of all the items in the order. It returns the result as a double

Answer:

1- In the abstract class *Item*, Identify the abstract method

```
public abstract void cost();
```

2- Write the Java code for the *Croissant*, *Bakery*, and the *Order* classes.

Croissant Class

```
public class Croissant extends Item {
private double baseprice;
private double typprice;
public Croissant (double p, double bp, double tp){
super(p);
bp = baseprice;
tp = typprice; }
public Croissant (Croissant cr) { }
public abstract void cost() {
    price = baseprice + typeprise; }

public void display() {
System.out.println(price + baseprice + typeprice);
}
```

4.25 / 01.25

0 comments Activity

Type a comment

College of Computer & Information Sciences
Information Technology Department

CSC 113 Final Exam

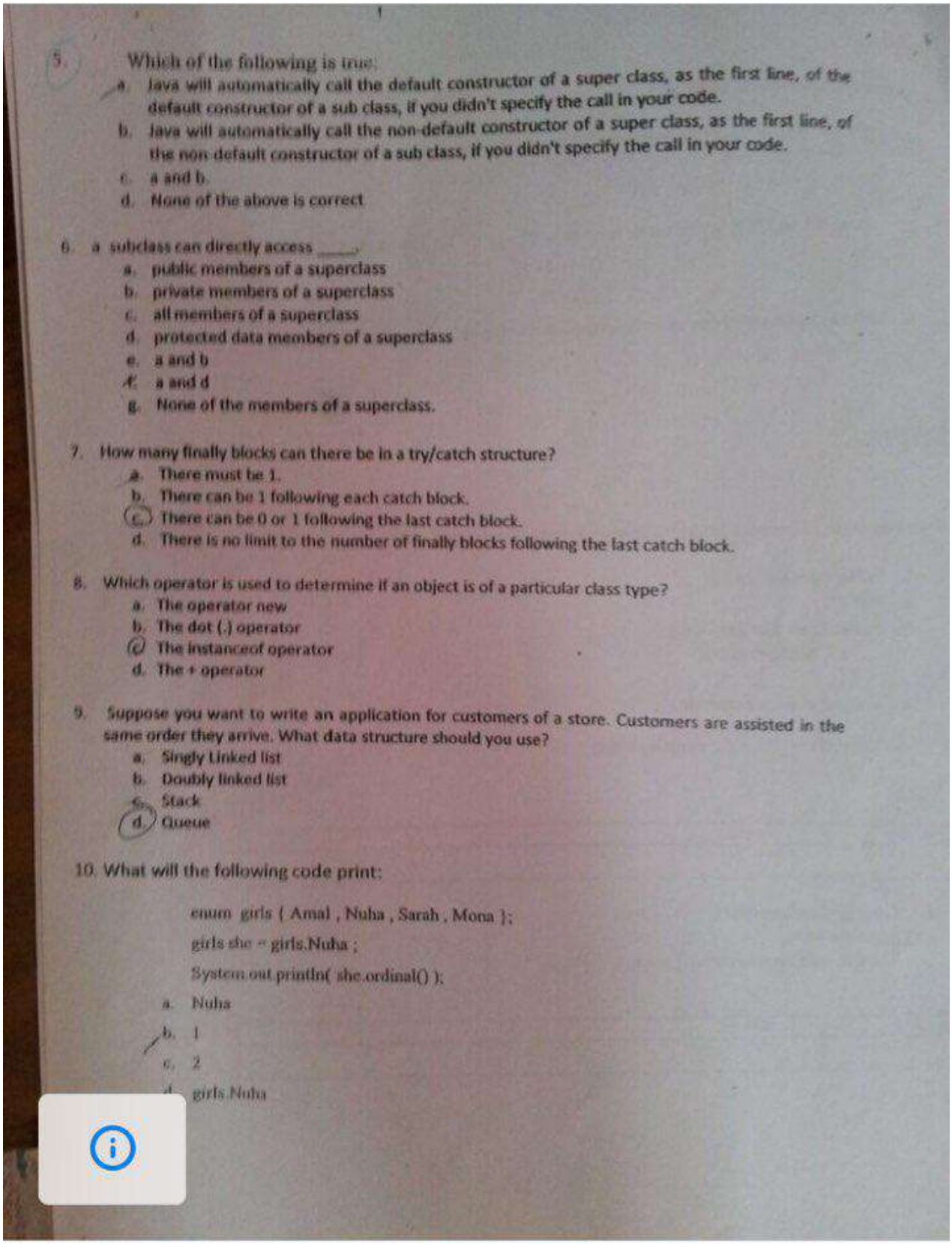
Write your name and ID in Arabic.

	الاسم:	الرقم الجامعي :
9205(8 - 9)	25194(10-11)	رقم الشعبه: ضعي دائرة

Question Number	Question points	Student Score
1	5	
2	6	
3	7	
4	3	
5	3	
6	3	
7	3	
8	10	
Total	40	

Question1: Multiple choice questions:

1. Which statement is true about an abstract class?
a. It cannot be instantiated.
b. It can be extended.
c. It will not have an abstract method.
d. It cannot have data members
e. a and b
f. a and d
g. None of the above



5. Which of the following is true:
   a. Java will automatically call the default constructor of a super class, as the first line, of the default constructor of a sub class, if you didn't specify the call in your code.
   b. Java will automatically call the non-default constructor of a super class, as the first line, of the non-default constructor of a sub class, if you didn't specify the call in your code.
   c. a and b
   d. None of the above is correct

6. a subclass can directly access ____.
   a. public members of a superclass
   b. private members of a superclass
   c. all members of a superclass
   d. protected data members of a superclass
   e. a and b
   f. a and d
   g. None of the members of a superclass.

7. How many finally blocks can there be in a try/catch structure?
   a. There must be 1.
   b. There can be 1 following each catch block.
   c. There can be 0 or 1 following the last catch block.
   d. There is no limit to the number of finally blocks following the last catch block.

8. Which operator is used to determine if an object is of a particular class type?
   a. The operator new
   b. The dot (.) operator
   c. The instanceof operator
   d. The + operator

9. Suppose you want to write an application for customers of a store. Customers are assisted in the same order they arrive. What data structure should you use?
   a. Singly Linked list
   b. Doubly linked list
   c. Stack
   d. Queue

10. What will the following code print:

```
enum girls { Amal , Nuha , Sarah , Mona };
girls she = girls.Nuha ;
System.out.println( she.ordinal() );
```

   a. Nuha
   b. 1
   c. 2
   d. girls.Nuha

0 comments Activity

Type a comment

2. Here is the hierarchy of exceptions related to array index and string index errors:

```
Exception
  +-- RuntimeException
        +-- IndexOutOfBoundsException
              +-- ArrayIndexOutOfBoundsException
              +-- StringIndexOutOfBoundsException
```

Suppose you had a method *X* that could throw both array index and string index exceptions. Which of the following statements are correct?

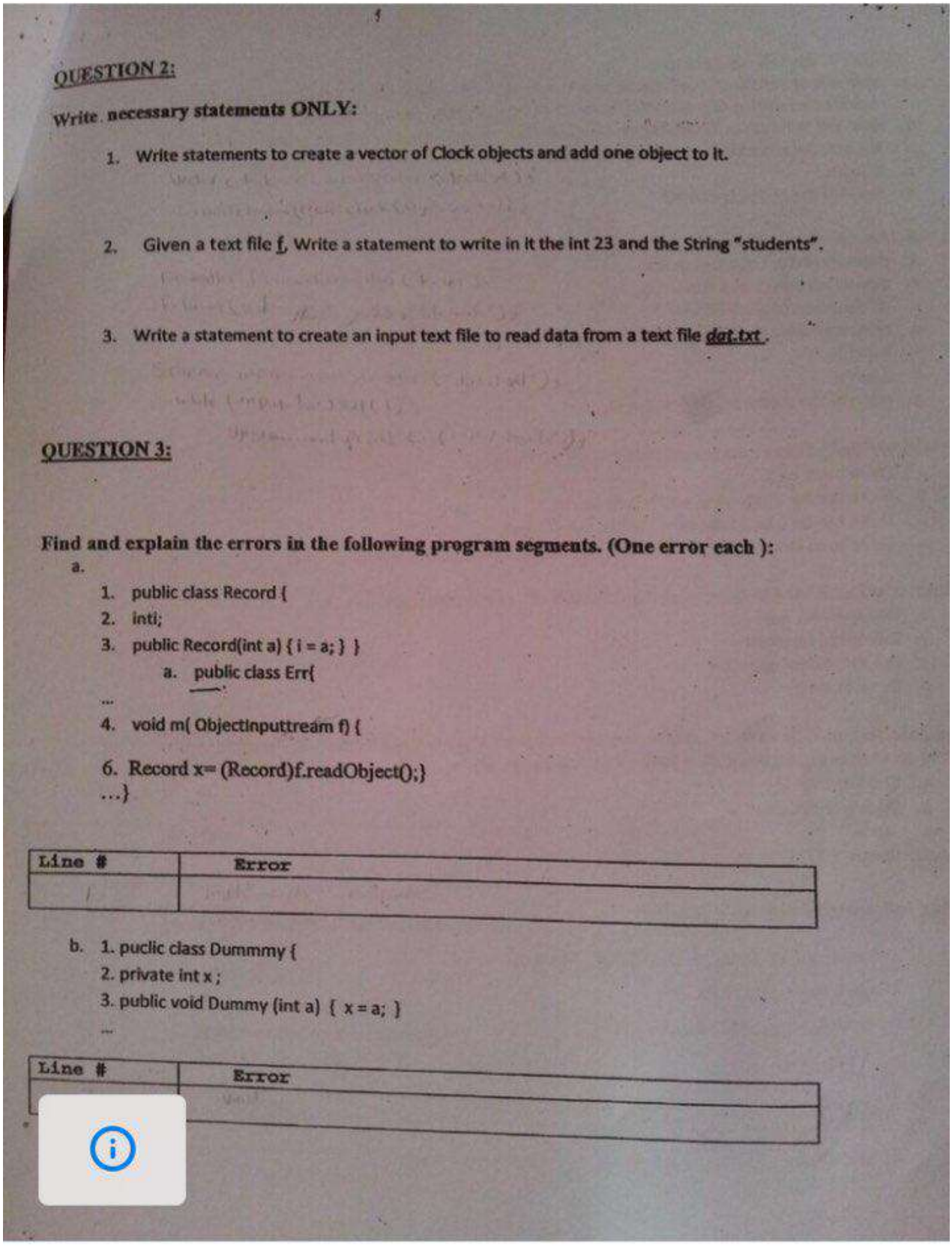
a. The declaration for *X* must include "`throws ArrayIndexOutOfBoundsException, StringIndexOutOfBoundsException`".
b. The declaration for *X* must include "throws `IndexOutOfBoundsException`".
c. The definition for *X* must include `try-catch` statements.
d. Any one of the above.
e. None of the above.

3. Which of the following can you perform using the `File` class?

a. Change the current directory.
b. Open a file for input.
c. Delete a file.
d. Get the absolute path.

4. A finally clause will execute

a. only if the try statement that precedes it does not throw an exception
b. only if the try statement that precedes it throws an exception that is caught
c. only if the try statement that precedes it throws an exception that is not caught
d. only if the try statement that precedes it throws an exception, whether it is caught or not
e. in any circumstance



QUESTION 2:

Write necessary statements ONLY:

1. Write statements to create a vector of Clock objects and add one object to it.

2. Given a text file f, Write a statement to write in it the int 23 and the String "students".

3. Write a statement to create an input text file to read data from a text file dat.txt .

QUESTION 3:

Find and explain the errors in the following program segments. (One error each):

a.

```
1. public class Record {
2. int i;
3. public Record(int a) { i = a; } }
      a. public class Err{
...
4. void m( ObjectInputtream f) {
6. Record x= (Record)f.readObject();}
...}
```

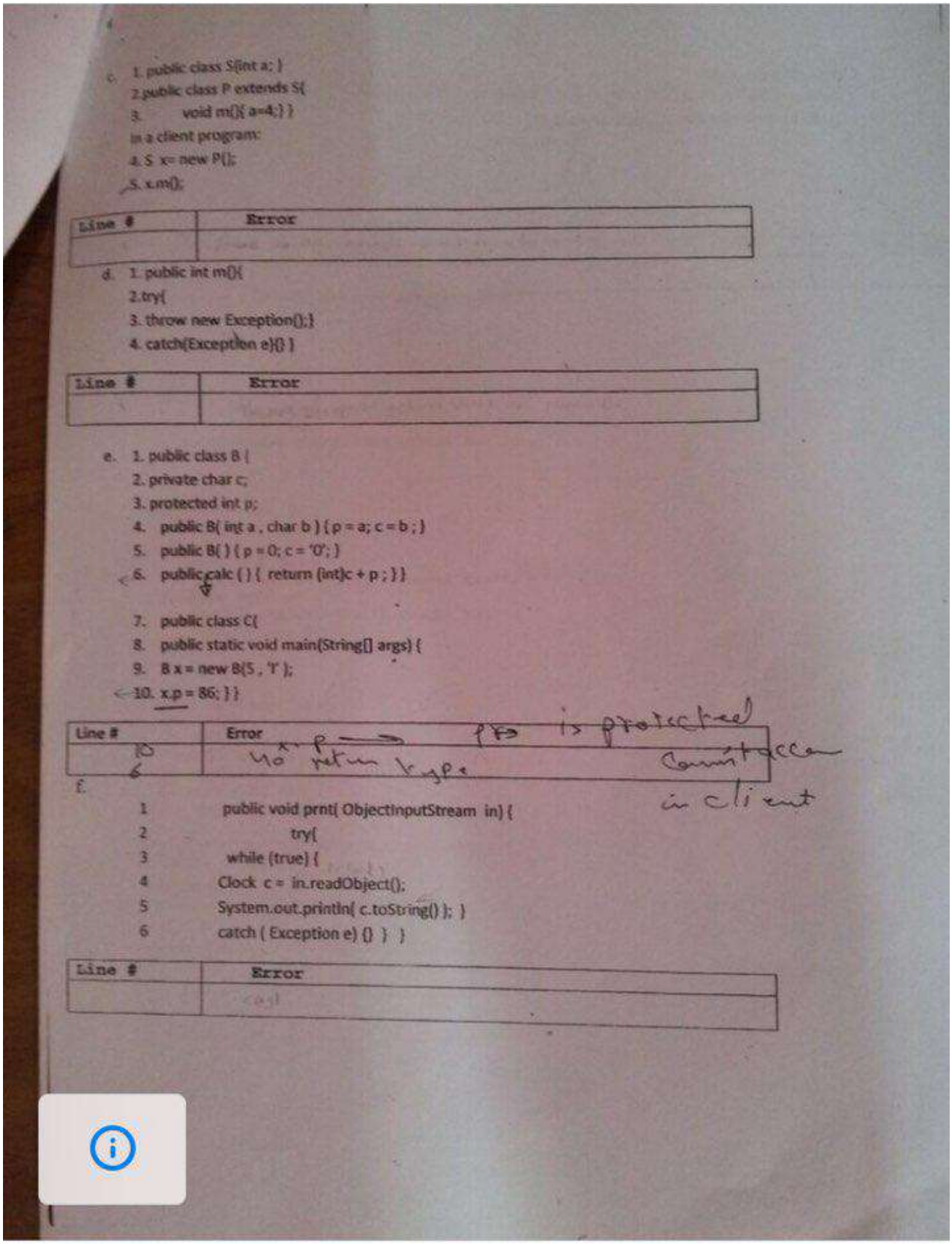
Line #	Error

b.

```
1. puclic class Dummmy {
2. private int x ;
3. public void Dummy (int a) { x = a; }
...
```

Line #	Error

0 comments Activity
Type a comment



c.
```
1. public class S{int a; }
2.public class P extends S{
3.        void m(){ a=4;} }
in a client program:
4. S x= new P();
5. x.m();
```

Line #	Error

d.
```
1. public int m(){
2.try{
3. throw new Exception();}
4. catch(Exception e){} }
```

Line #	Error

e.
```
1. public class B {
2. private char c;
3. protected int p;
4.   public B( int a , char b ) { p = a; c = b ; }
5.   public B( ) { p = 0; c = '0'; }
6.   public calc ( ) { return (int)c + p ; } }

7.   public class C{
8.   public static void main(String[] args) {
9.   B x = new B(5 , 'f' );
10. x.p = 86; } }
```

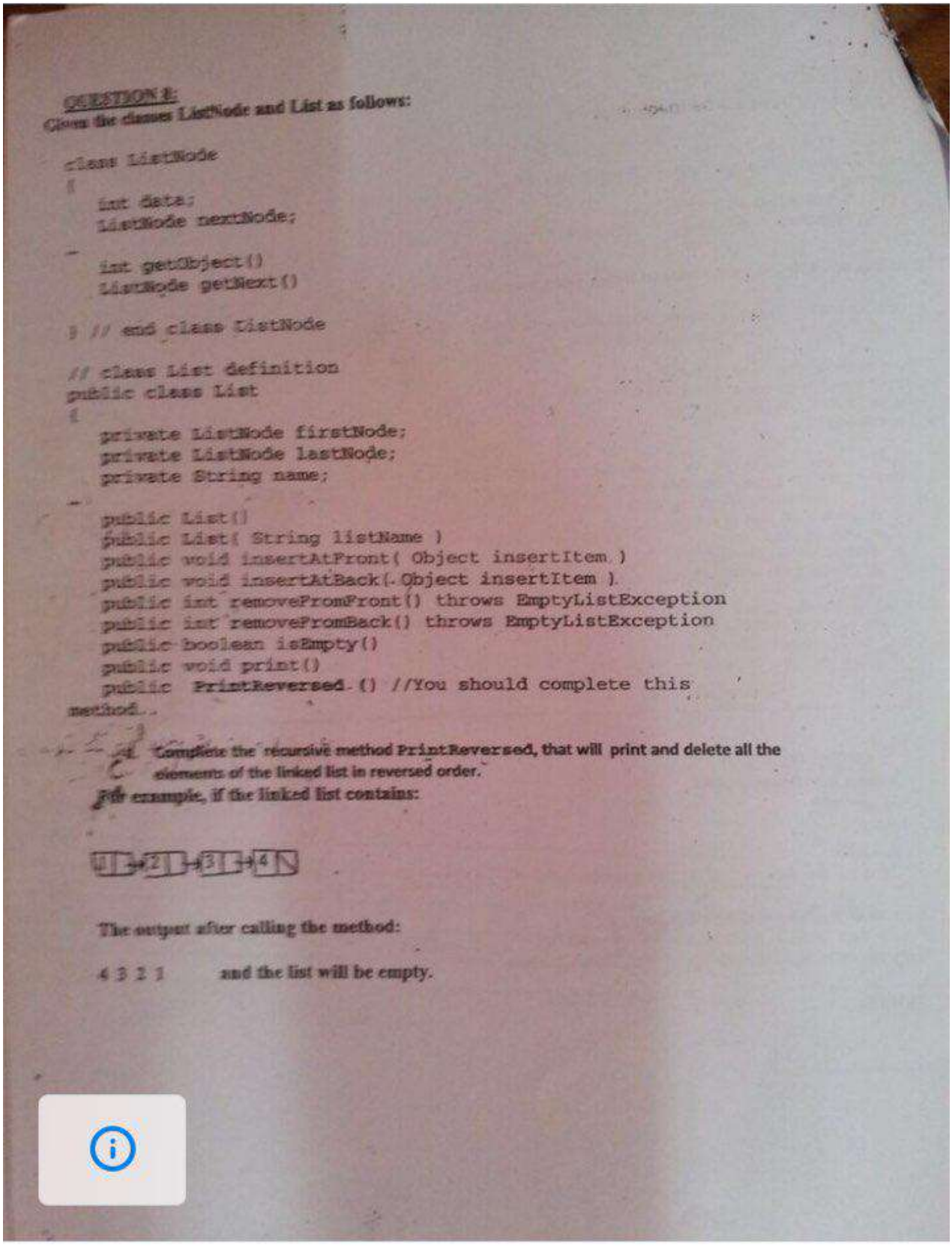
Line #	Error
10	x.p → p is protected
6	No return type. Constructor in client

f.
```
1          public void prnt( ObjectInputStream  in) {
2                  try{
3           while (true) {
4          Clock  c =  in.readObject();
5          System.out.println( c.toString() ); }
6          catch ( Exception e) {} } }
```

Line #	Error

0 comments Activity

Type a comment



QUESTION 1:
Given the classes ListNode and List as follows:

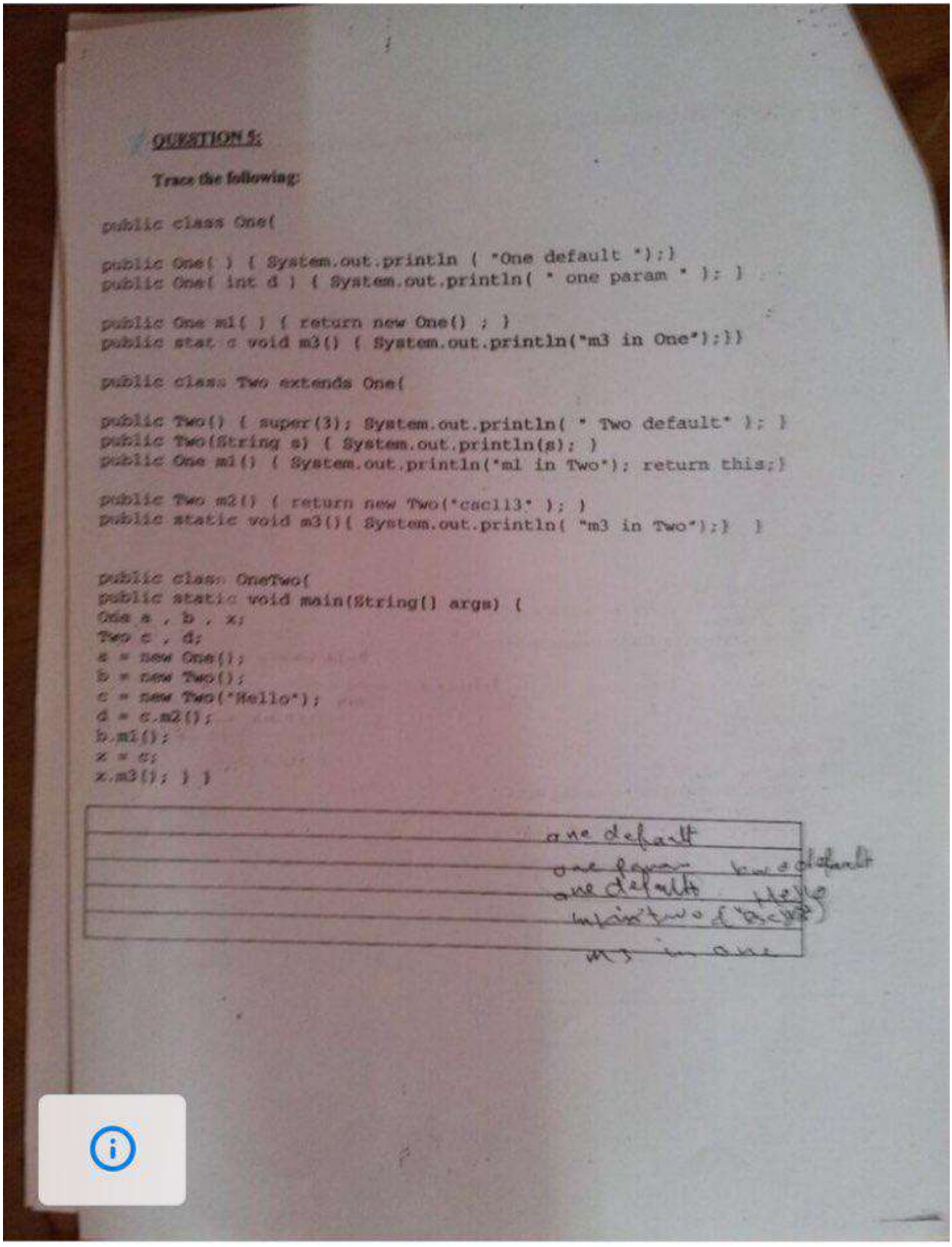
```
class ListNode
{
   int data;
   ListNode nextNode;

   int getObject()
   ListNode getNext()

} // end class ListNode

// class List definition
public class List
{
   private ListNode firstNode;
   private ListNode lastNode;
   private String name;

   public List()
   public List( String listName )
   public void insertAtFront( Object insertItem )
   public void insertAtBack( Object insertItem )
   public int removeFromFront() throws EmptyListException
   public int removeFromBack() throws EmptyListException
   public boolean isEmpty()
   public void print()
   public  PrintReversed () //You should complete this
method..
```

Complete the recursive method PrintReversed, that will print and delete all the elements of the linked list in reversed order.

For example, if the linked list contains:

The output after calling the method:

4 3 2 1 and the list will be empty.



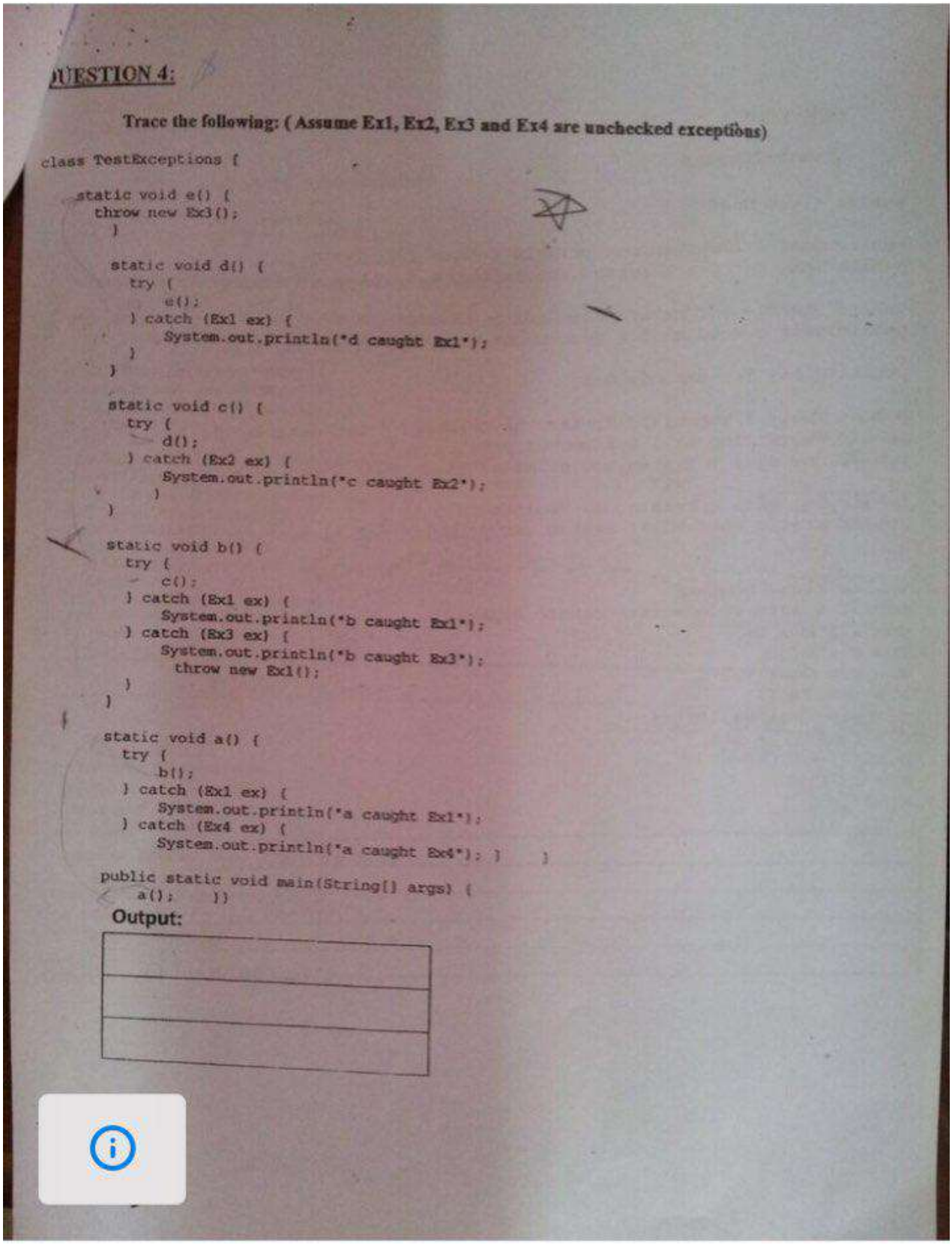
QUESTION 5:

Trace the following:

```
public class One{

public One( ) { System.out.println ( "One default ");}
public One( int d ) { System.out.println( " one param " ); }

public One m1( ) { return new One() ; }
public stat c void m3() { System.out.println("m3 in One");}}

public class Two extends One{

public Two() { super(3); System.out.println( " Two default" ); }
public Two(String s) { System.out.println(s); }
public One m1() { System.out.println("m1 in Two"); return this;}

public Two m2() { return new Two("cscl13" ); }
public static void m3(){ System.out.println( "m3 in Two");}  }


public class OneTwo{
public static void main(String[] args) {
One a , b , x;
Two c , d;
a = new One();
b = new Two();
c = new Two("Hello");
d = c.m2();
b.m1();
x = c;
x.m3(); } }
```

one default
one param two default
one default Hello
m1 in two cscl13
m3 in one

0 comments Activity
Type a comment

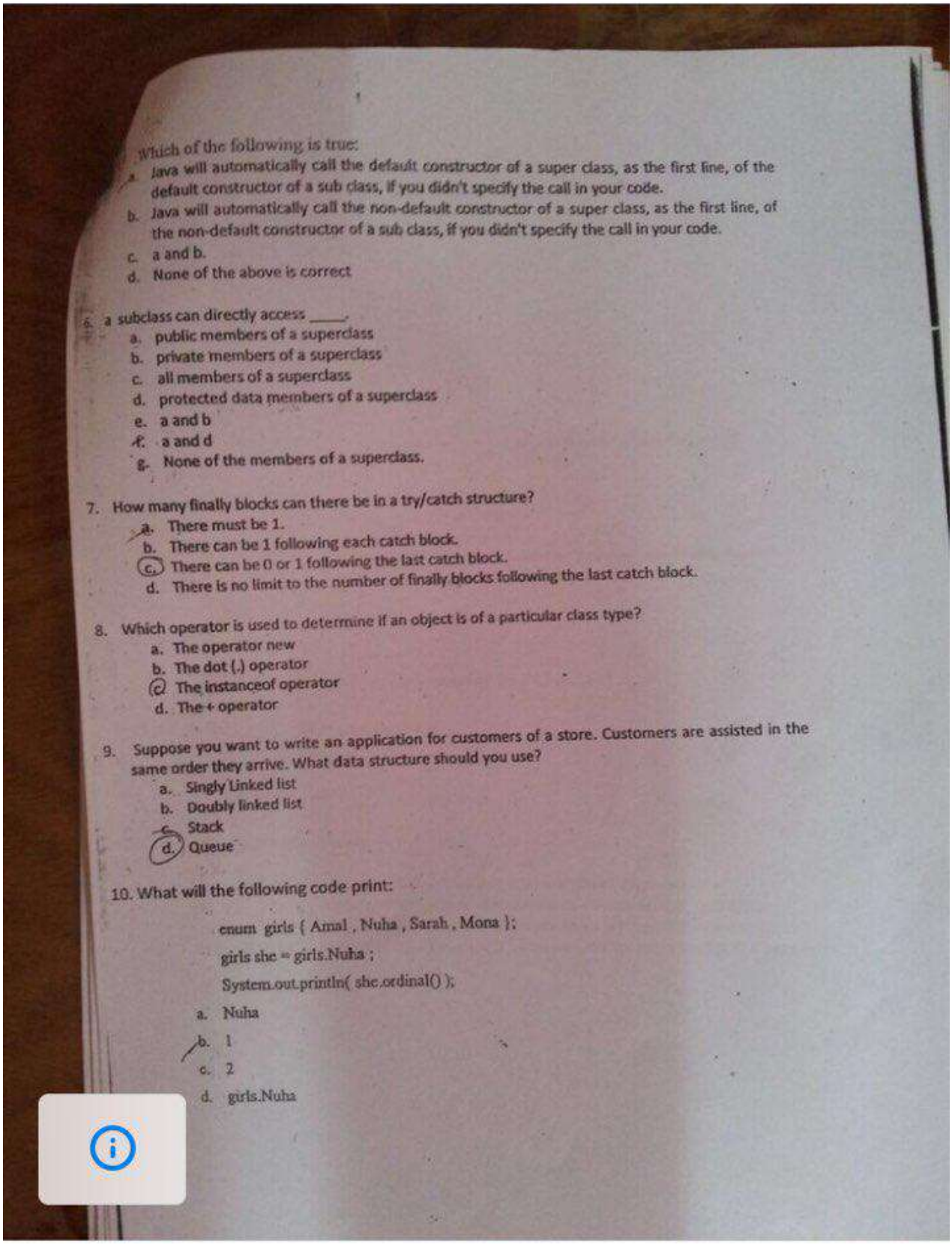


QUESTION 4:

Trace the following: (Assume Ex1, Ex2, Ex3 and Ex4 are unchecked exceptions)

```
class TestExceptions {

   static void e() {
     throw new Ex3();
    }

    static void d() {
      try {
          e();
      } catch (Ex1 ex) {
          System.out.println("d caught Ex1");
      }
    }

    static void c() {
      try {
          d();
      } catch (Ex2 ex) {
          System.out.println("c caught Ex2");
      }
    }

    static void b() {
      try {
          c();
      } catch (Ex1 ex) {
          System.out.println("b caught Ex1");
      } catch (Ex3 ex) {
          System.out.println("b caught Ex3");
           throw new Ex1();
      }
    }

    static void a() {
      try {
          b();
      } catch (Ex1 ex) {
          System.out.println("a caught Ex1");
      } catch (Ex4 ex) {
          System.out.println("a caught Ex4"); }    }

    public static void main(String[] args) {
       a();     }}
```

Output:



Which of the following is true:
a. Java will automatically call the default constructor of a super class, as the first line, of the default constructor of a sub class, if you didn't specify the call in your code.
b. Java will automatically call the non-default constructor of a super class, as the first line, of the non-default constructor of a sub class, if you didn't specify the call in your code.
c. a and b.
d. None of the above is correct

6. a subclass can directly access ____.
a. public members of a superclass
b. private members of a superclass
c. all members of a superclass
d. protected data members of a superclass
e. a and b
f. a and d
g. None of the members of a superclass.

7. How many finally blocks can there be in a try/catch structure?
a. There must be 1.
b. There can be 1 following each catch block.
c. There can be 0 or 1 following the last catch block.
d. There is no limit to the number of finally blocks following the last catch block.

8. Which operator is used to determine if an object is of a particular class type?
a. The operator new
b. The dot (.) operator
c. The instanceof operator
d. The + operator

9. Suppose you want to write an application for customers of a store. Customers are assisted in the same order they arrive. What data structure should you use?
a. Singly Linked list
b. Doubly linked list
c. Stack
d. Queue

10. What will the following code print:

```
enum girls { Amal , Nuha , Sarah , Mona };
girls she = girls.Nuha ;
System.out.println( she.ordinal() );
```

a. Nuha
b. 1
c. 2
d. girls.Nuha

0 comments Activity

Type a comment