

American University of Sharjah
School of Engineering
Department of Computer Engineering
P. O. Box 26666
Sharjah, UAE



Instructor: Imran A. Zualkernan
Office: ESB-2066
Phone: 971-6-515 2953
Fax: 971-6-515 2979
e-mail: izualkernan@aus.edu
Semester: Fall 2020

Name: Amir Mohideen
ID : 74559

80/100

COE312 – Mid Term Exam

Fall 2020

Time: 1 hour and 30 minutes

Score: /100

The quiz open book and notes. Please paste screenshot of the output with your solution. Attempt all questions.

Q1. (50 points) Create three types of exceptions *IgnoreException*, *SoftException* and *HardException*.

IgnoreException is a *Green* card, *SoftException* is a *Green* and a *Yellow* card while the *HardException* is a *Red* card.

- *Green* cards need to specify the reason for why the card was issued.
- A *Yellow* card needs to specify the reason as well as the conditions under which the card can be issued and the conditions under which card cannot be issued.
- The *Red* card specifies the reason and the associated penalty.

Show how your classes work on the following program. **You are NOT allowed to change anything in the program.**

```
public class Driver {  
    public static void main(String[] args) throws Exception {  
        try {  
            throw new IgnoreException("no worries mate!");  
        } catch (Exception e) {  
            System.out.println(e);  
        }  
        try {  
            throw new SoftException("no worries mate!", "may be this was not a good  
idea", "standing", "sitting");  
        } catch (Exception e) {  
            System.out.println(e);  
        }  
        try {  
            throw new HardException("this is not acceptable", "do 10 situps");  
        } catch (Exception e) {  
            System.out.println(e);  
        }  
    }  
}
```

```
}
```

Expected Program Output:

This exception must be ignored and it is green because no worries mate!

This exception can potentially be ignored and it applies when standing and does not apply when sitting, and it is also green because no worries mate!

[java.lang.Exception](#): This exception must be addressed because this is not acceptable and the penalty is 'do 10 situps'

at HardException.toString(HardException.java:22)

at java.base/java.lang.String.valueOf(String.java:3042)

at java.base/java.io.PrintStream.println(PrintStream.java:897)

at Driver.main(Driver.java:24)

Solution (Past your code here):

Unformatted code will result in automatic deduction of 10 points.

//class IgnoreException that implements greencard

public class IgnoreException **extends** Exception **implements** GreenCard{

String str;

//non default constructor

public IgnoreException(String string) {
 // **TODO** Auto-generated constructor stub
 super(string);
 str = string; ?
}

// default constructor

public IgnoreException() {
 // **TODO** Auto-generated constructor stub
 super();
}

//prints the exception message as required

public String getMessage() {

 return "This exception must be ignored "+
gcolour()+ " because " + **this**.str;
}

//interface method implementation

public String gcolour() {
 // **TODO** Auto-generated method stub
 return "it is green"; ?
}

}

public class SoftException **extends** Exception **implements**
GreenCard, YellowCard {

String s1, s2, s3, s4;

// default constructor

public SoftException() {

// **TODO** Auto-generated constructor stub

s1 = "";

s2 = "";

s3 = "";

s4 = "";

}

// non default constructor

public SoftException(String str1, String str2, String str3, String
str4) {

// **TODO** Auto-generated constructor stub

s1 = str1;

s2 = str2;

s3 = str3;

s4 = str4;

}

// prints the exception message as required

public String getMessage() {

return "This exception can potentially be ignored and it
applies when "

+ **this**.s3 + " and does not apply when " + **this**.s4 + "
and " + gcolour() + s1;

}

// interface method implementation

public String gcolour() {

// **TODO** Auto-generated method stub

return "it is also green";

}

// interface method implementation

public String ycolour() {

Not correct

```

    }
    // TODO Auto-generated method stub
    return "it is yellow";
}
}

```

NOT

Correct!

```

public class HardException extends Exception implements
RedCard {

```

```

    String s1;
    String s2;

```

```

//default constructor

```

```

public HardException() {
    // TODO Auto-generated constructor stub
    //super();
    s1 = "";
    s2 = "";
}

```

```

//non default constructor

```

```

public HardException(String str1, String str2) {
    // TODO Auto-generated constructor stub
    //super(string);
    s1=str1;
    s2=str2;
}

```

```

//prints the exception message as required

```

```

public String getMessage() {

```

```

    return "This exception must be addressed because "+
s1 +" and the penalty is "+ s2 + "";

```

```

}

```

```

public String rcolour() {

```

```

    // TODO Auto-generated method stub
    return "it is red";
}

```

```

}

```

```

public interface GreenCard {
    String gcolour();
}

```

7 ?

public interface YellowCard {

String ycolour();

}

public interface RedCard {

String rcolour();

}

Need
better
methods

Screenshot (Paste your screenshot here)

Leaving this space empty will result in automatic deduction of 10 points.

-10

<terminated> main (3) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_261.jdk/Contents/Home/bin/java (Oct 24, 2020 2:00:40 PM - 2:00:41 PM)
[IgnoreException](#): This exception must be ignored it is green because no worries mate!
[SoftException](#): This exception can potentially be ignored and it applies when standing and does not apply when sitting and it is also greenno worries mate!
[HardException](#): This exception must be addressed because this is not acceptable and the penalty is 'do 10 situps'

1. WRONG METHODS
2. NO TO STRING
3. NOT ALLOW FOR RED

Q2. (50 points) Write two classes called *CrazyInputStream* and *CrazyOutputStream* with the following functionality.

CrazyOutputStream implements the following two functions:

- *writeDouble(double d)*
 - writes the double *d* in IEEE format followed by a randomly generated double number to any output stream.
- *writeFloat(float f)*
 - writes the double *d* in IEEE format followed by a randomly generated float number to any output stream.

CrazyInputStream implements the following two functions:

- *double readDouble()*
 - reads a double in IEEE format and the next double in IEEE format from any input stream and returns the product of two numbers.
- *float readFloat(float f)*
 - reads a float in IEEE format and the next float in IEEE from any input stream and returns the sum of two numbers.

Show how your classes work using the following sample program. **You are not allowed to change the program.**

```
import java.io.FileInputStream;
import java.io.FileOutputStream;

public class Driver {

    public static void main(String[] args) throws Exception {

        CrazyOutputStream cout = new CrazyOutputStream(new
        FileOutputStream("funny.txt"));

        cout.writeDouble(3);
        cout.writeFloat((float)4);

        cout.close();

        CrazyInputStream cin = new CrazyInputStream(new FileInputStream("funny.txt"));

        System.out.println(cin.readDouble());
        System.out.println(cin.readFloat());

        cin.close();

    }
}
```

Program Expected Output:

```
0.30231531152552815
4.98682
```



The file funny.txt is 24 bytes long. The actual numbers may vary every time because random numbers are being used.

Solution (Paste your code here):

Unformatted code will result in automatic deduction of 10 points.

```
import java.io.*;
import java.io.DataOutputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;

import java.util.Random;

public class CrazyOutputStream {

    //non default constructor
    public CrazyOutputStream(FileOutputStream fout)
    {
        DataOutputStream dout = new DataOutputStream(fout);
    }

    //writes the double d in IEEE format followed by a randomly
    generated double number to any output stream.
    void writeDouble(double d)
    {

        dout.writeDouble(d);

        Random rand = new Random();
        double rand_dub1 = rand.nextDouble();
        dout.writeDouble(rand_dub1);
    }
}
```

//writes the float f in IEEE format followed by a randomly generated float number to any output stream.

```

void writeFloat(float f)
{
    dout.writeFloat(f);
    Random rand = new Random();
    float rand_dub1 = (float) rand.nextDouble();
    dout.writeDouble(rand_dub1);
}

```

```

void close()
{
}

```

?

```

import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;

```

```

public class CrazyInputStream {

```

```

    //non default constructor

```

```

    public CrazyInputStream(FileInputStream fin)
    {
        DataInputStream din = new DataInputStream(fin);
    }

```

```

    //function that reads two double from data stream, and returns
    the product of the numbers in IEEE format

```

```

    public String readDouble()
    {
        double x = din.readFloat();
        double y = din.readFloat();

        return ((x * y) + "\n");
    }

```

WRONG

```

    //reads a float in IEEE format and the next float in IEEE from
    any input stream and returns the sum of two numbers.

```

```

    public String readFloat()
    {
        float x = din.readFloat();
        float y = din.readFloat();
        return (x + y + "\n");
    }

```

WRONG


```
}
```

```
void close()
```

```
{
```

```
}
```

```
}
```

CLOSE IO STREAM

Screenshot (Paste your code screenshot here):

Leaving this space empty will result in automatic deduction of 10 points.

<terminated> Driver (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_261.jdk/Contents/Home/bin/java

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

dout cannot be resolved

dout cannot be resolved

at CrazyOutputStream.writeDouble(CrazyOutputStream.java:23)

at Driver.main(Driver.java:10)

OUTPUT

1. NO CLOSE
2. ~~NO~~ STRING ID
Double.
3. NOT COMPILES

10