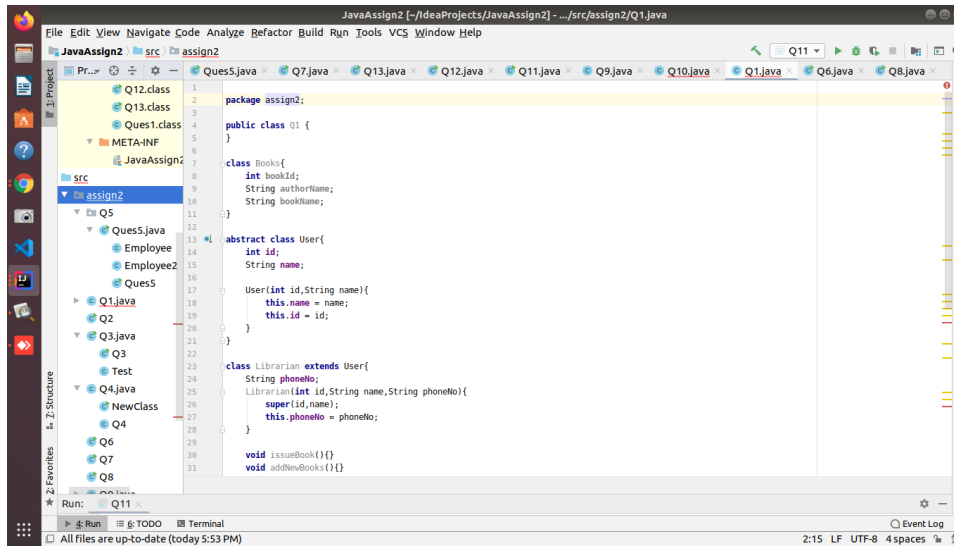


Question: 1:

Create Java classes having suitable attributes for Library management system. Use OOPs concepts in your design. Also try to use interfaces and abstract classes.



```
package assign2;

public class Q1 {
}

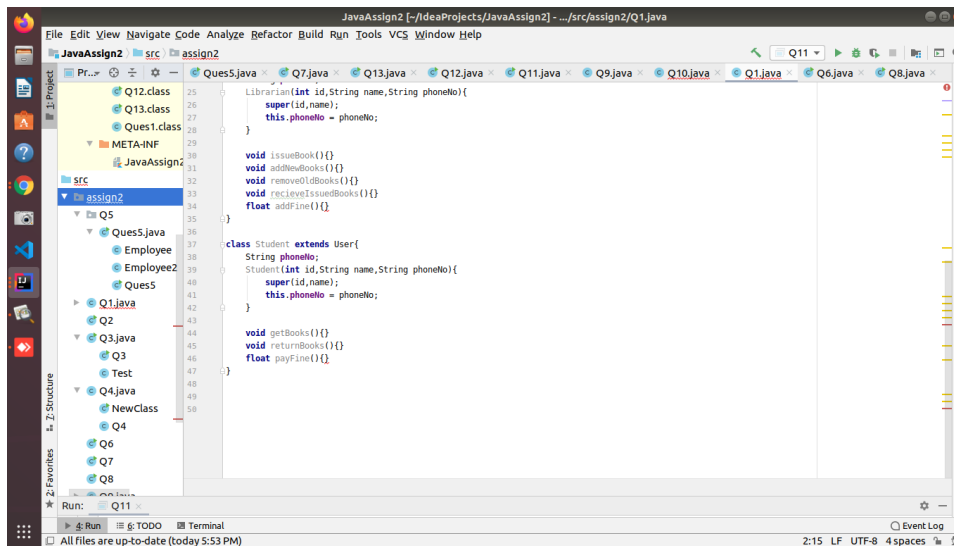
class Books {
    int bookId;
    String authorName;
    String bookName;
}

abstract class User {
    int id;
    String name;

    User(int id, String name) {
        this.name = name;
        this.id = id;
    }
}

class Librarian extends User {
    String phoneno;
    Librarian(int id, String name, String phoneno) {
        super(id, name);
        this.phoneno = phoneno;
    }

    void issueBook() {}
    void addNewBooks() {}
}
```



```
Librarian(int id, String name, String phoneno) {
    super(id, name);
    this.phoneno = phoneno;
}

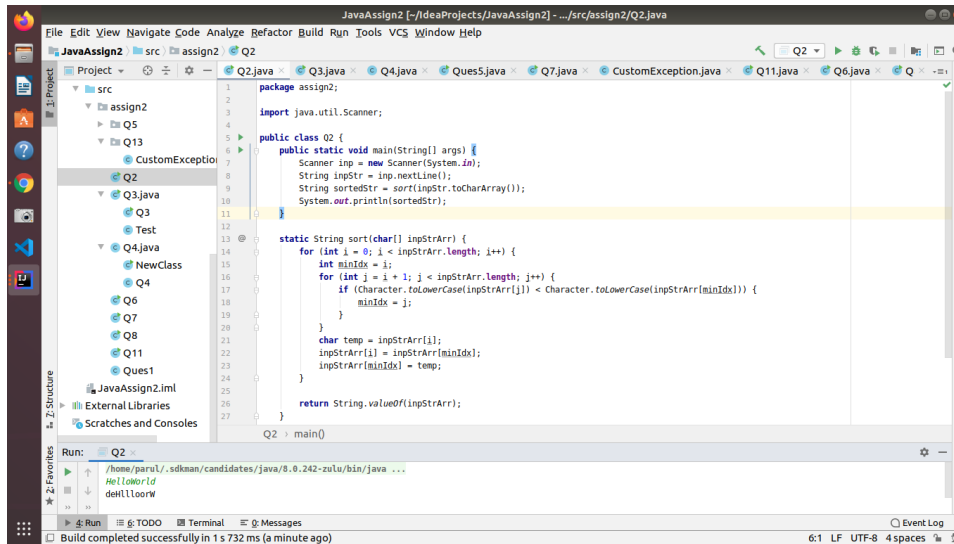
void issueBook() {}
void addNewBooks() {}
void removeOldBooks() {}
void receiveIssuedBooks() {}
float addFine() {}
}

class Student extends User {
    String phoneno;
    Student(int id, String name, String phoneno) {
        super(id, name);
        this.phoneno = phoneno;
    }

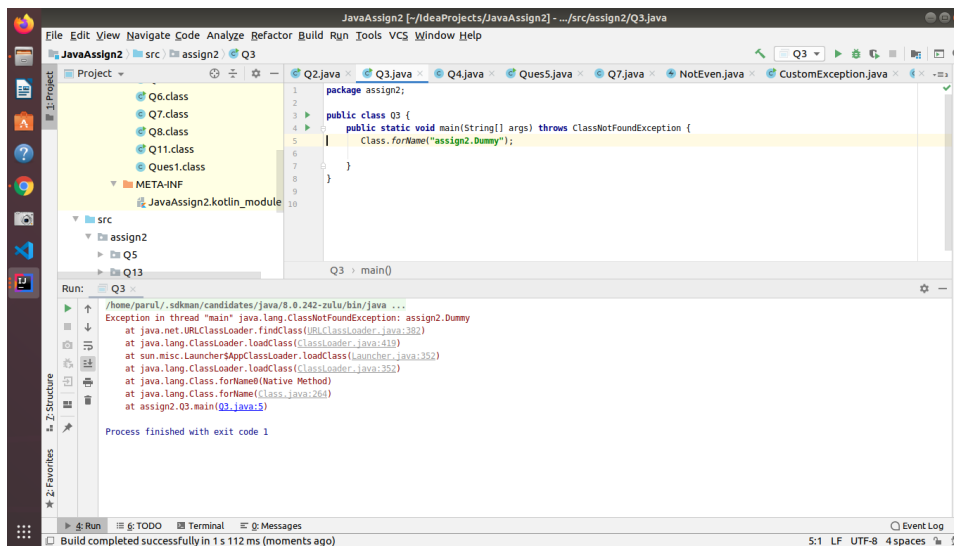
    void getBooks() {}
    void returnBooks() {}
    float payFine() {}
}
```

Question: 2:

WAP to sorting string without using string Methods?



Question: 3:
WAP to produce NoClassDefFoundError and ClassNotFoundException exception.



The screenshot shows the IntelliJ IDEA IDE with a project named 'JavaAssign2'. The 'src' directory contains a package 'assign2' with several Java files: Q1.class, Q2.java, Q3.java, Q4.java, Q5, Q13, Q2, Q3.java, Q4.java, and Q6. The 'Q3.java' file is open in the editor, showing the following code:

```
1 package assign2;
2
3 public class Q3 {
4     public static void main(String[] args) throws ClassNotFoundException {
5         Test t = new Test();
6     }
7
8
9     class Test{
10
11     }
12 }
```

The 'Run' tab at the bottom shows the execution of 'Q3' with the following error message:

```
Exception in thread "main" java.lang.NoClassDefFoundError: assign2/Test
    at assign2.Q3.main(Q3.java:3)
Caused by: java.lang.ClassNotFoundException: assign2.Test
    at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:419)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:352)
    ... 1 more
Process finished with exit code 1
```

Question: 4:
WAP to create singleton class.

The screenshot shows the IntelliJ IDEA IDE with a project named 'JavaAssign2'. The 'src' directory contains a package 'assign2' with several Java files: Q2.java, Q3.java, Q4.java, Q5.java, Q7.java, CustomException.java, Q11.java, Q6.java, and Q8.java. The 'Q4.java' file is open in the editor, showing the following code:

```
1 package assign2;
2
3 public class Q4{
4     private static Q4 instance = null;
5
6     public int i;
7
8     private Q4() { this.i = 4; }
9
10
11     public static Q4 getInstance(){
12         if(instance==null){
13             instance = new Q4();
14         }
15         return instance;
16     }
17 }
18
19
20 class NewClass {
21     public static void main(String[] args) {
22         Q4 obj = Q4.getInstance();
23         Q4 obj1 = Q4.getInstance();
24         Q4 obj2 = Q4.getInstance();
25     }
26 }
```

The 'Run' tab at the bottom shows the execution of 'NewClass' with the following error message:

```
Exception in thread "main" java.lang.NoClassDefFoundError: assign2/Q4
    at NewClass.main(NewClass.java:22)
Caused by: java.lang.ClassNotFoundException: assign2.Q4
    at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:419)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:352)
    ... 1 more
Process finished with exit code 1
```



```
1 package assign2;
2
3 import java.util.InputMismatchException;
4 import java.util.Scanner;
5
6 public class Q6 {
7     public static void main(String[] args) {
8         Scanner inp = new Scanner(System.in);
9         try{
10             int inpInt = inp.nextInt();
11             int res = 100/inpInt;
12             System.out.println(res);
13         } catch(InputMismatchException e){
14             System.out.println("Enter Valid Integer");
15             e.printStackTrace();
16         } catch (ArithmeticException e){
17             System.out.println("Arithmetic Exception " + e.getMessage());
18             e.printStackTrace();
19         } finally {
20             System.out.println("Finally");
21         }
22     }
23 }
24
```

Run: Q6

Event Log

All Files are up-to-date (a minute ago)

```
1 package assign2;
2
3 import java.util.InputMismatchException;
4 import java.util.Scanner;
5
6 public class Q6 {
7     public static void main(String[] args) {
8         Scanner inp = new Scanner(System.in);
9         try{
10             int inpInt = inp.nextInt();
11             int res = 100/inpInt;
12             System.out.println(res);
13         } catch(InputMismatchException e){
14             System.out.println("Enter Valid Integer");
15             e.printStackTrace();
16         } catch (ArithmeticException e){
17             System.out.println("Arithmetic Exception " + e.getMessage());
18             e.printStackTrace();
19         } finally {
20             System.out.println("Finally");
21         }
22     }
23 }
24
```

Run: Q6

Event Log

All Files are up-to-date (a minute ago)

```
1 package assign2;
2
3 import java.util.InputMismatchException;
4 import java.util.Scanner;
5
6 public class Q6 {
7     public static void main(String[] args) {
8         Scanner inp = new Scanner(System.in);
9         try{
10             int inpInt = inp.nextInt();
11             int res = 100/inpInt;
12             System.out.println(res);
13         } catch(InputMismatchException e){
14             System.out.println("Enter Valid Integer");
15             e.printStackTrace();
16         } catch (ArithmeticException e){
17             System.out.println("Arithmetic Exception " + e.getMessage());
18             e.printStackTrace();
19         } finally {
20             System.out.println("Finally");
21         }
22     }
23 }
24
```

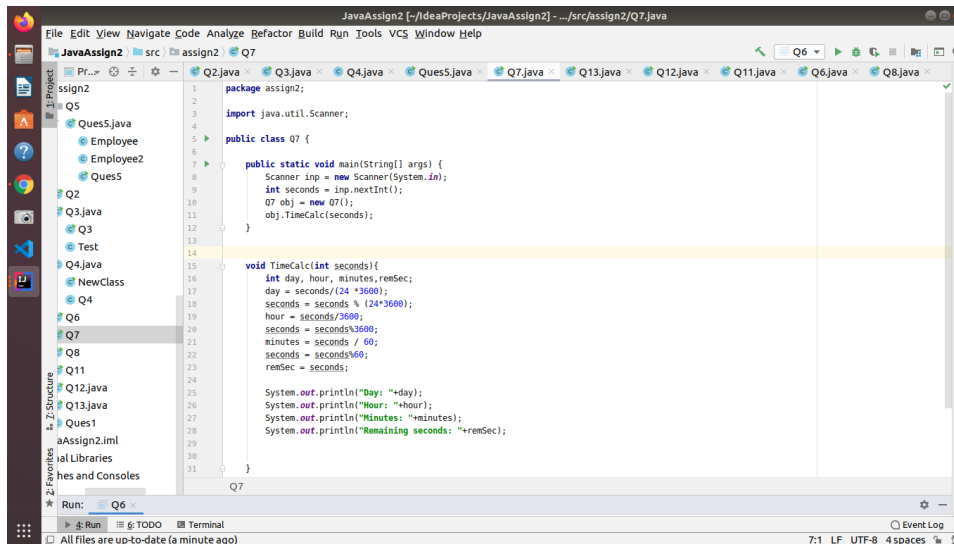
Run: Q6

Event Log

All Files are up-to-date (a minute ago)

Question: 7:

WAP to convert seconds into days, hours, minutes and seconds.



```
package assign2;

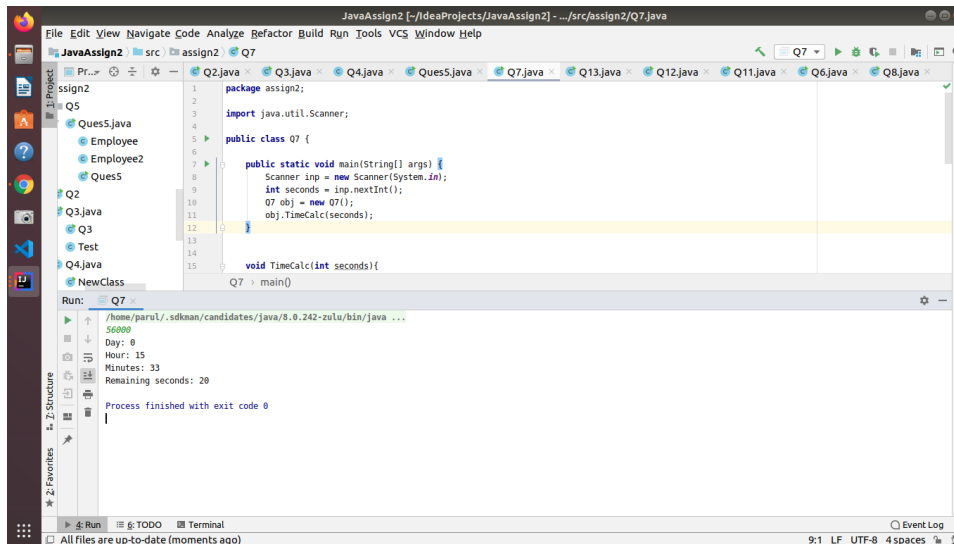
import java.util.Scanner;

public class Q7 {

    public static void main(String[] args) {
        Scanner inp = new Scanner(System.in);
        int seconds = inp.nextInt();
        Q7 obj = new Q7();
        obj.TimeCalc(seconds);
    }

    void TimeCalc(int seconds){
        int day, hour, minutes, remSec;
        day = seconds/(24*3600);
        seconds = seconds % (24*3600);
        hour = seconds/3600;
        seconds = seconds%3600;
        minutes = seconds / 60;
        seconds = seconds%60;
        remSec = seconds;

        System.out.println("Day: "+day);
        System.out.println("Hour: "+hour);
        System.out.println("Minutes: "+minutes);
        System.out.println("Remaining seconds: "+remSec);
    }
}
```



```
Run: Q7
/home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
5000
Day: 0
Hour: 15
Minutes: 33
Remaining seconds: 20
Process finished with exit code 0
```

Question: 8:

WAP to read words from the keyboard until the word done is entered. For each word except done, report whether its first character is equal to its last character. For the required loop, use a)while statement b)do-while statement

```

package assign2;

import java.util.Scanner;

public class Q8 {
    public static void main(String[] args) {
        Scanner inp = new Scanner(System.in);
        String inpStr = "";
        while(!inpStr.equals("done")){
            inpStr = inp.nextLine();
            if(inpStr.charAt(inpStr.length()-1)!=inpStr.charAt(0)){
                System.out.println("First and Last character are same");
            }
            System.out.println("Do while loop starts");
            inpStr = "";
            do{
                inpStr = inp.nextLine();
                if(inpStr.charAt(inpStr.length()-1)!=inpStr.charAt(0)) {
                    System.out.println("First and Last character are same");
                }
            }while (!inpStr.equals("done"));
        }
    }
}

```

```

Run: Q8
/home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
hi
hello
bob
First and Last character are same
Do while loop starts
world
bob
First and Last character are same
done
First and Last character are same
Do while loop starts
hi
bob
First and Last character are same
done
Process finished with exit code 0

```

Question: 9:

Design classes having attributes for furniture where there are wooden chairs and tables, metal chairs and tables. There are stress and fire tests for each products.


```

package assign2;

public class Q9 {

    abstract class Furniture{
        String material;
        boolean isFireProof;
        int stressResistance;

        Furniture(String material){
            this.material=material;
        }

        void FireTest(boolean isFireProof){
            this.isFireProof=isFireProof;
        }

        void StressResistance(int stressResistance){
            this.stressResistance =stressResistance;
        }
    }

    class Chair extends Furniture{
        float height;
        float width;
        Chair(String material, float height, float width){
            super(material);
            this.height=height;
            this.width=width;
        }
    }
}

```

```

package assign2;

public class Q10 {

    abstract class Furniture{
        String material;
        boolean isFireProof;
        int stressResistance;

        Furniture(String material){
            this.material=material;
        }

        void FireTest(boolean isFireProof){
            this.isFireProof=isFireProof;
        }

        void StressResistance(int stressResistance){
            this.stressResistance =stressResistance;
        }
    }

    class Chair extends Furniture{
        float height;
        float width;
        Chair(String material, float height, float width){
            super(material);
            this.height=height;
            this.width=width;
        }
    }

    class Table extends Furniture{
        float height;
        float width;
        Table(String material, float height, float width){
            super(material);
            this.height=height;
            this.width=width;
        }
    }
}

```

Question: 10:

Design classes having attributes and method(only skeleton) for a coffee shop. There are three different actors in our scenario and i have listed the different actions they do also below.

* Customer

- Pays the cash to the cashier and places his order, get a token number back
 - Waits for the intimation that order for his token is ready
 - Upon intimation/notification he collects the coffee and enjoys his drink
- (Assumption: Customer waits till the coffee is done, he wont timeout and cancel the order. Customer always likes the drink served. Exceptions like he not liking his coffee, he getting wrong coffee are not considered to keep the design simple.)

* Cashier

- Takes an order and payment from the customer
- Upon payment, creates an order and places it into the order queue

- Intimates the customer that he has to wait for his token and gives him his token
(Assumption: Token returned to the customer is the order id. Order queue is unlimited. With a simple modification, we can design for a limited queue size)

* Barista

- Gets the next order from the queue
- Prepares the coffee
- Places the coffee in the completed order queue
- Places a notification that order for token is ready

```
package assign2;

public class Q10 {

    class Customer{
        String customerName;
        String address;

        Customer(String customerName, String address){
            this.customerName = customerName;
            this.address = address;
        }

        void makeOrder(){

        }

        void requestForBill(){

        }

        void payBill(){

        }

    }

    class Cashier{
        String cashierName;
        Cashier(String cashierName){
            this.cashierName = cashierName;
        }

        void takeOrder(){

        }

        void generateBill(){

        }

        int acceptPayment(){

        }

        void addOrderToOrderQueue(){

        }

    }

    class Barista{
        int id;
        String name;
        Barista(int id, String name){
            this.id = id;
            this.name = name;
        }

        void prepareCoffee(){

        }

    }

    Order o;
    Cashier c;
    Barista b;

    public static void main(String[] args) {
        Order o = new Order();
        Cashier c = new Cashier("Cashier");
        Barista b = new Barista(1, "Barista");
        o = c.takeOrder();
        c.generateBill();
        c.acceptPayment();
        c.addOrderToOrderQueue();
        b.prepareCoffee();
    }
}
```

```
package assign2;

public class Q10 {

    class Customer{
        String customerName;
        String address;

        Customer(String customerName, String address){
            this.customerName = customerName;
            this.address = address;
        }

        void makeOrder(){

        }

        void requestForBill(){

        }

        void payBill(){

        }

    }

    class Cashier{
        String cashierName;
        Cashier(String cashierName){
            this.cashierName = cashierName;
        }

        void takeOrder(){

        }

        void generateBill(){

        }

        int acceptPayment(){

        }

        void addOrderToOrderQueue(){

        }

    }

    class Barista{
        int id;
        String name;
        Barista(int id, String name){
            this.id = id;
            this.name = name;
        }

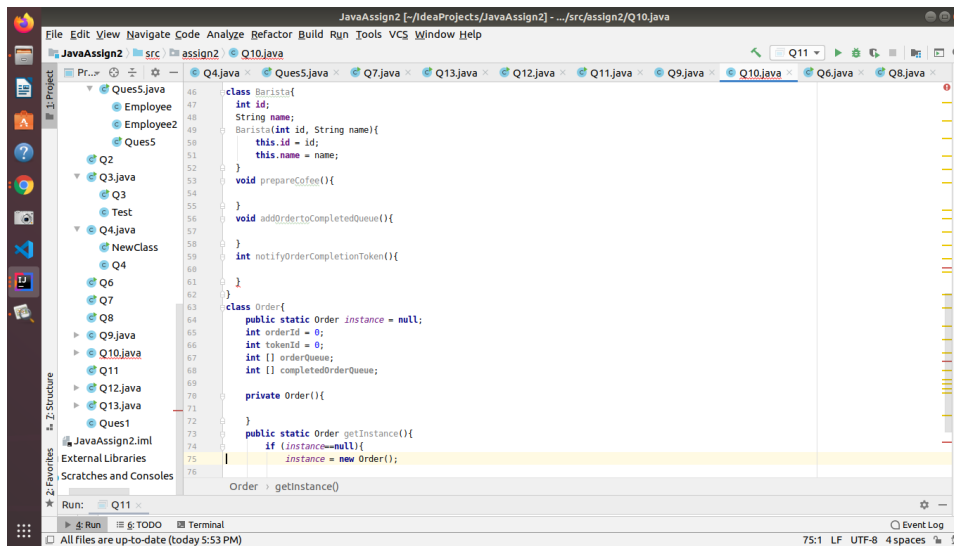
        void prepareCoffee(){

        }

    }

    Order o;
    Cashier c;
    Barista b;

    public static void main(String[] args) {
        Order o = new Order();
        Cashier c = new Cashier("Cashier");
        Barista b = new Barista(1, "Barista");
        o = c.takeOrder();
        c.generateBill();
        c.acceptPayment();
        c.addOrderToOrderQueue();
        b.prepareCoffee();
    }
}
```



Question: 11:

Convert the following code so that it uses nested while statements instead of for statements:

```

int s = 0;
int t = 1;
for (int i = 0; i < 10; i++)
{
    s = s + i;
    for (int j = i; j > 0; j--)
    {
        t = t * (j - i);
    }
    s = s * t;
    System.out.println("T is " + t);
}
System.out.println("S is " + s);

```

```

package assign2;

public class Q11 {
    public static void main(String[] args) {
        int s = 0;
        int t = 1;
        int i = 0;
        while (i < 10) {
            s = s + i;
            int j = i;
            while (j > 0) {
                t = t * (j - i);
                j--;
            }
            s = s * t;
            System.out.println("T is " + t);
            t++;
            System.out.println("S is " + s);
        }
    }
}

```

```

Run: Q11
/home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
T is 1
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
S is 0
Process finished with exit code 0

```

Question: 12:

What will be the output on new Child(); ?

class Parent extends Grandparent {

```

{
    System.out.println("instance - parent");
}

```

```

public Parent() {
    System.out.println("constructor - parent");
}

```

```

static {
    System.out.println("static - parent");
}

```

```

}
}

class Grandparent {
static {
System.out.println("static – grandparent");
}

{
System.out.println("instance - grandparent");
}

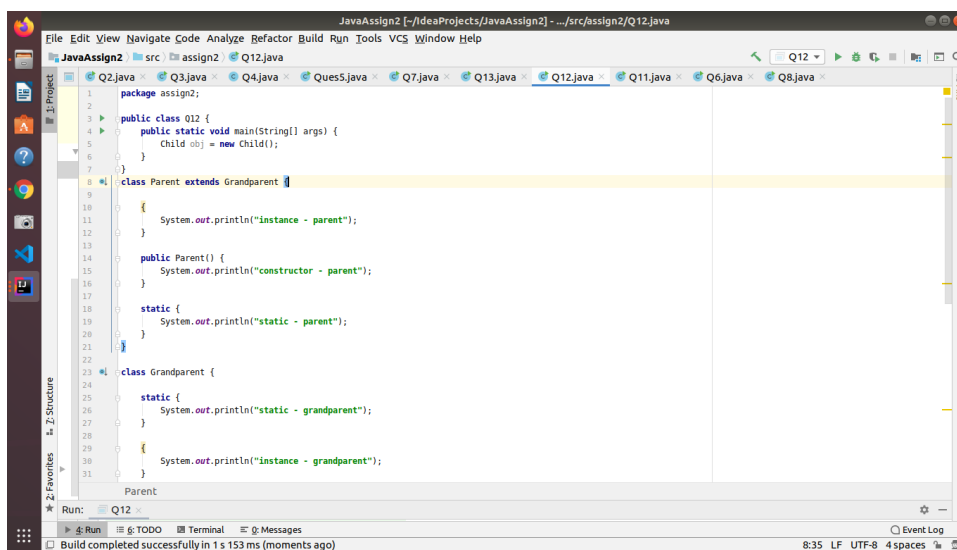
public Grandparent() {
System.out.println("constructor - grandparent");
}
}

class Child extends Parent {

public Child() {
System.out.println("constructor - child");
}

static {
System.out.println("static - child");
}
}
}
System.out.println("instance - child");
}
}
}

```



```
28
29
30 System.out.println("instance - grandparent");
31
32
33 public Grandparent() {
34     System.out.println("constructor - grandparent");
35 }
36
37
38 class Child extends Parent {
39
40     public Child() {
41         System.out.println("constructor - child");
42     }
43
44     static {
45         System.out.println("static - child");
46     }
47
48     public void main() {
49         System.out.println("instance - child");
50     }
51
52
53 }
```

Run: Q12

Build completed successfully in 1 s 153 ms (a minute ago)

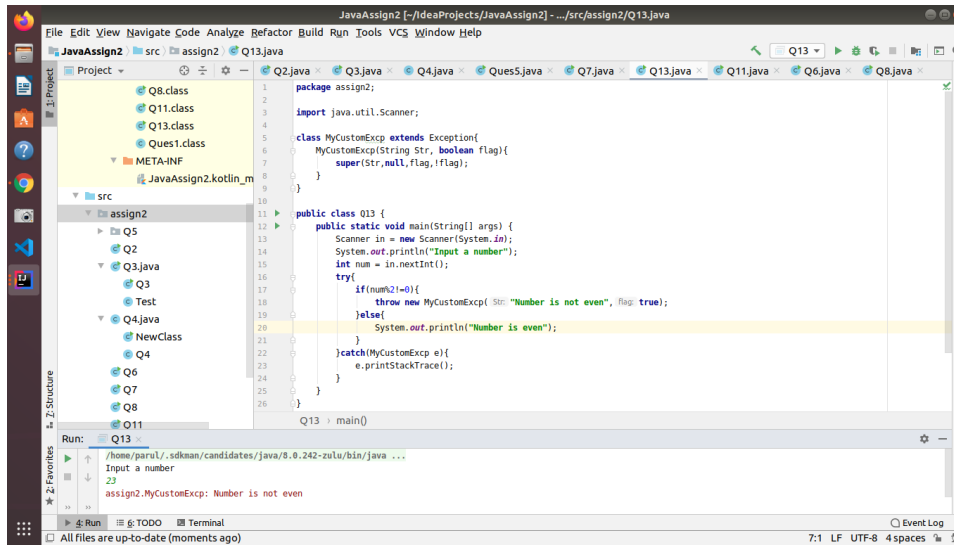
```
Run: Q12
/home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
static - grandparent
static - parent
static - child
instance - grandparent
constructor - grandparent
instance - parent
constructor - parent
instance - child
constructor - child

Process finished with exit code 0
```

Build completed successfully in 1 s 153 ms (a minute ago)

Question: 13:

Create a custom exception that do not have any stack trace.



Question: 15: