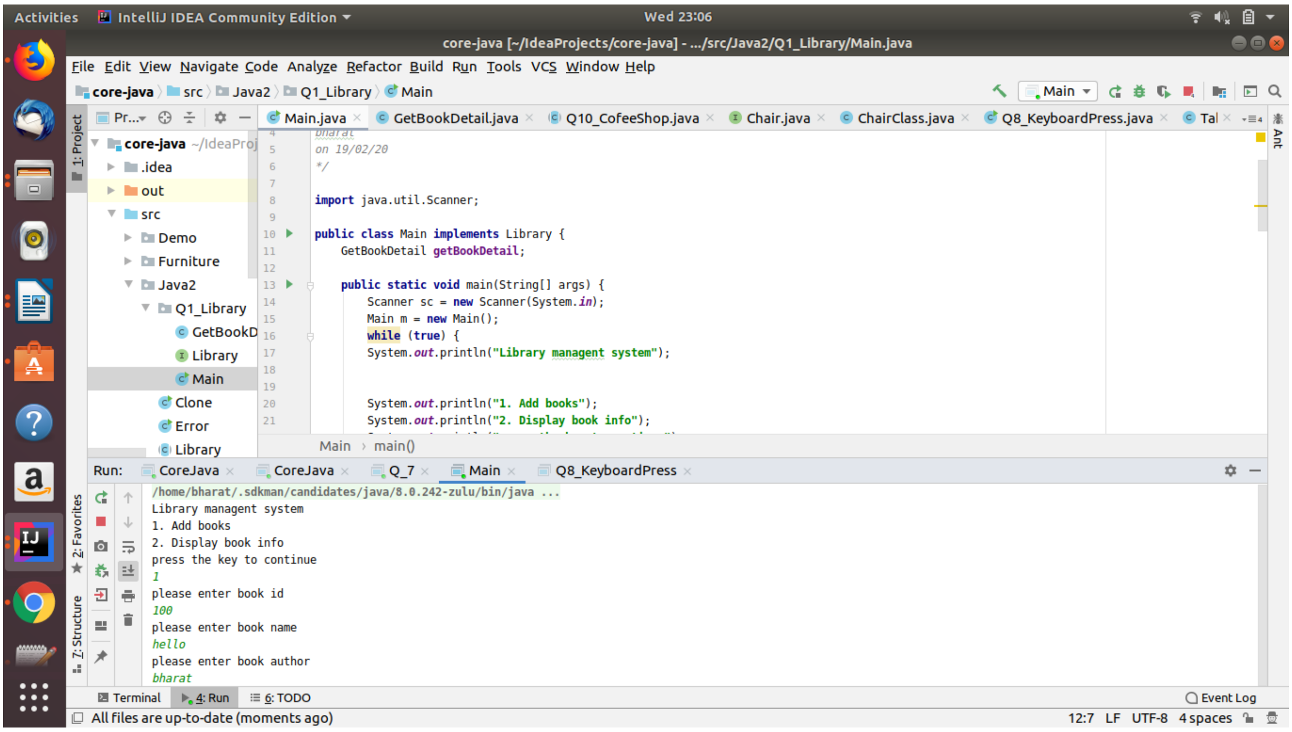
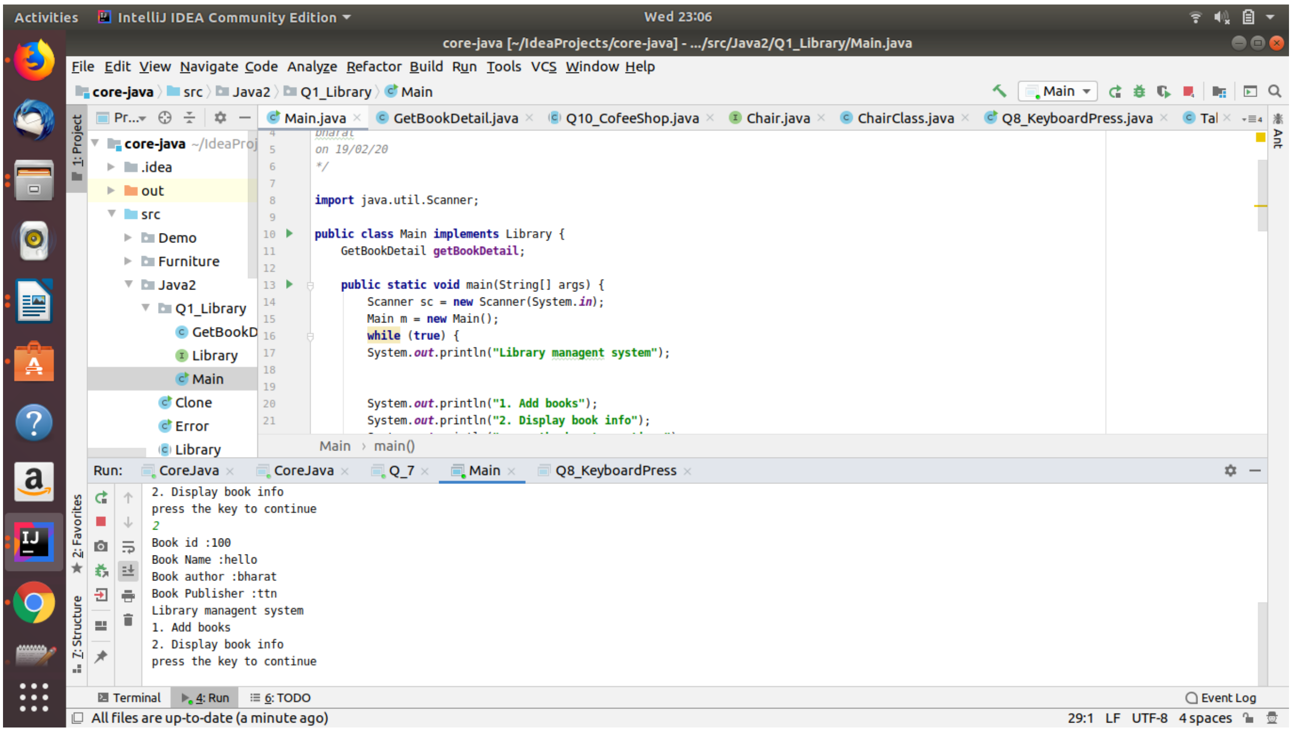
Q.1 Create Java classes having suitable attributes for Library management system.Use OOPs concepts in your design.Also try to use interfaces and abstract classes.

Code reference(GetBookDetail.java,Library.java,Main.java)



```
bharat
on 19/02/20
*/

import java.util.Scanner;

public class Main implements Library {
    GetBookDetail getBookDetail;

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Main m = new Main();
        while (true) {
        System.out.println("Library managent system");


        System.out.println("1. Add books");
        System.out.println("2. Display book info");
```

```
/home/bharat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Library managent system
1. Add books
2. Display book info
press the key to continue
1
please enter book id
100
please enter book name
hello
please enter book author
bharat
```

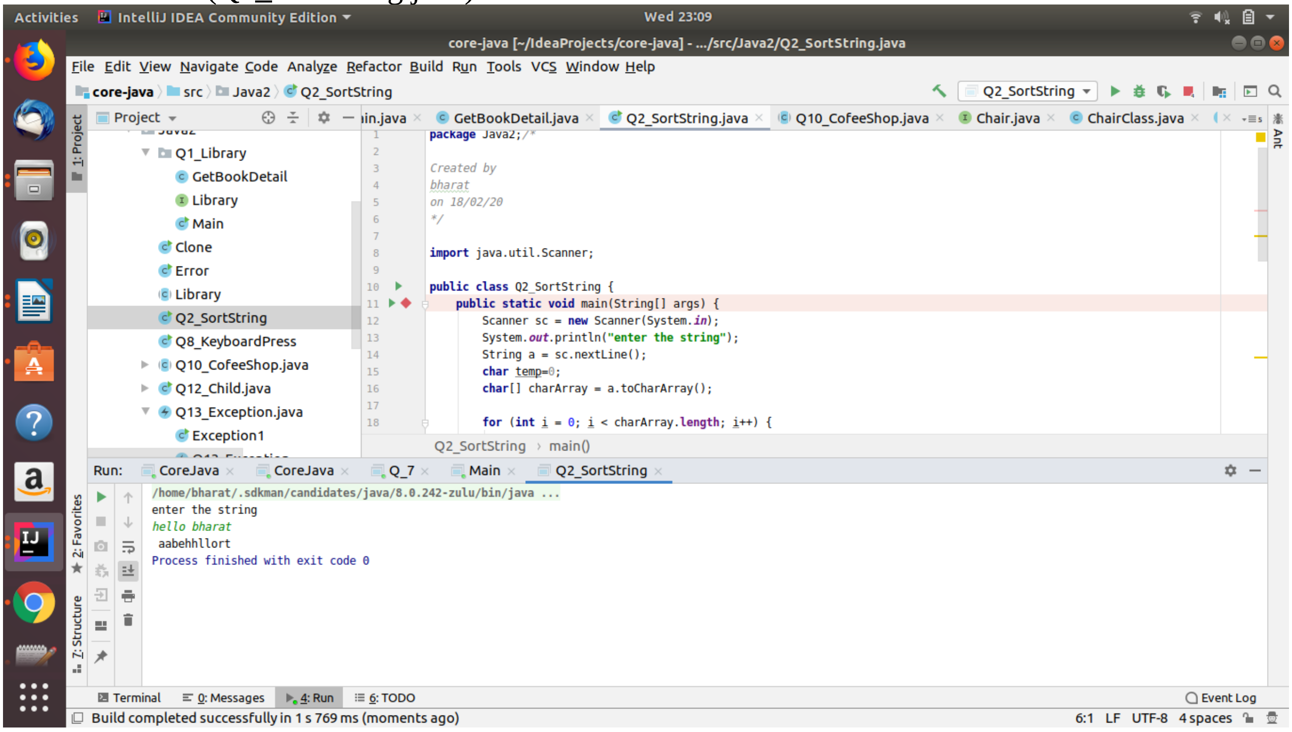


```
bharat
on 19/02/20
*/

import java.util.Scanner;

public class Main implements Library {
    GetBookDetail getBookDetail;

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Main m = new Main();
        while (true) {
        System.out.println("Library managent system");


        System.out.println("1. Add books");
        System.out.println("2. Display book info");
```

```
2. Display book info
press the key to continue
2
Book id :100
Book Name :hello
Book author :bharat
Book Publisher :ttn
Library managent system
1. Add books
2. Display book info
press the key to continue
```

Q.2 WAP to sorting string without using string Methods?.

Code reference (Q2_SortString.java)



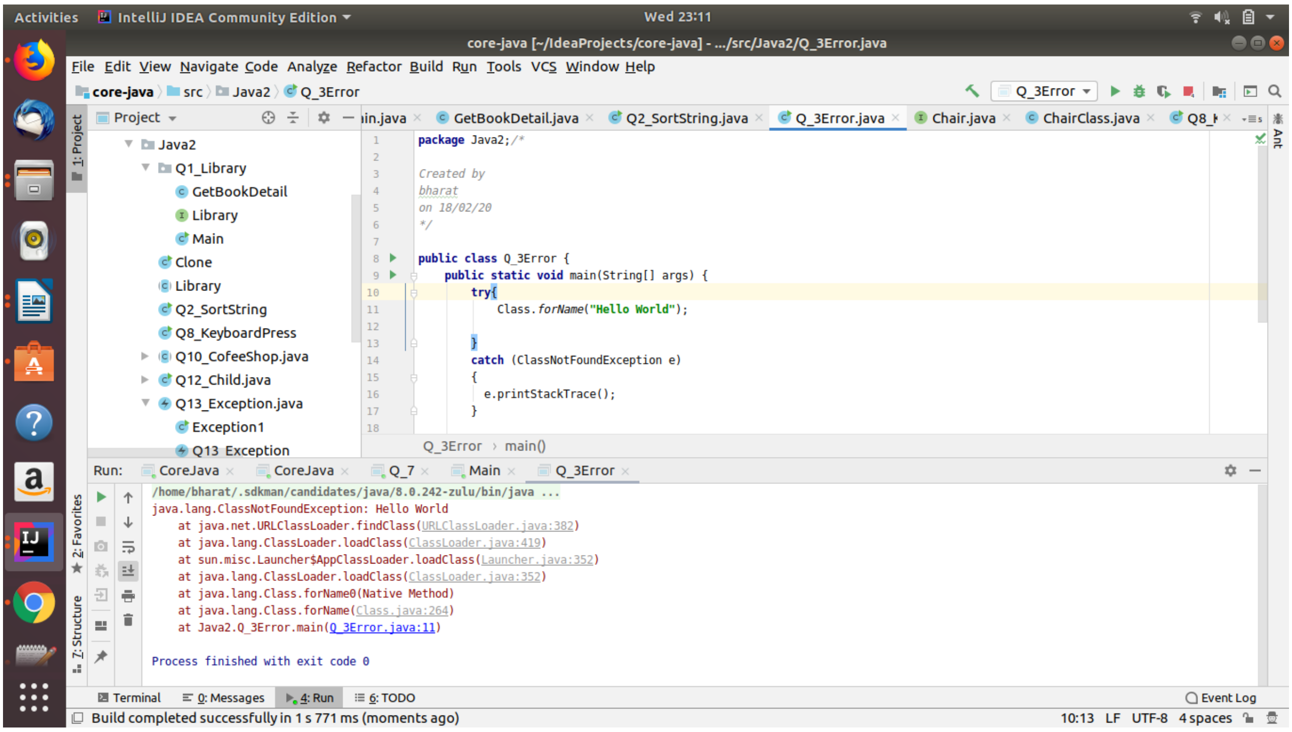
```
package Java2;/*

Created by
bharat
on 18/02/20
*/

import java.util.Scanner;

public class Q2_SortString {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter the string");
        String a = sc.nextLine();
        char temp=0;
        char[] charArray = a.toCharArray();

        for (int i = 0; i < charArray.length; i++) {
```

```
/home/bharat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
enter the string
hello bharat
 aabehhllort
Process finished with exit code 0
```

Build completed successfully in 1 s 769 ms (moments ago)

Q.3 WAP to produce NoClassDefFoundError and ClassNotFoundException exception.

Code reference(Q_3Error.java)



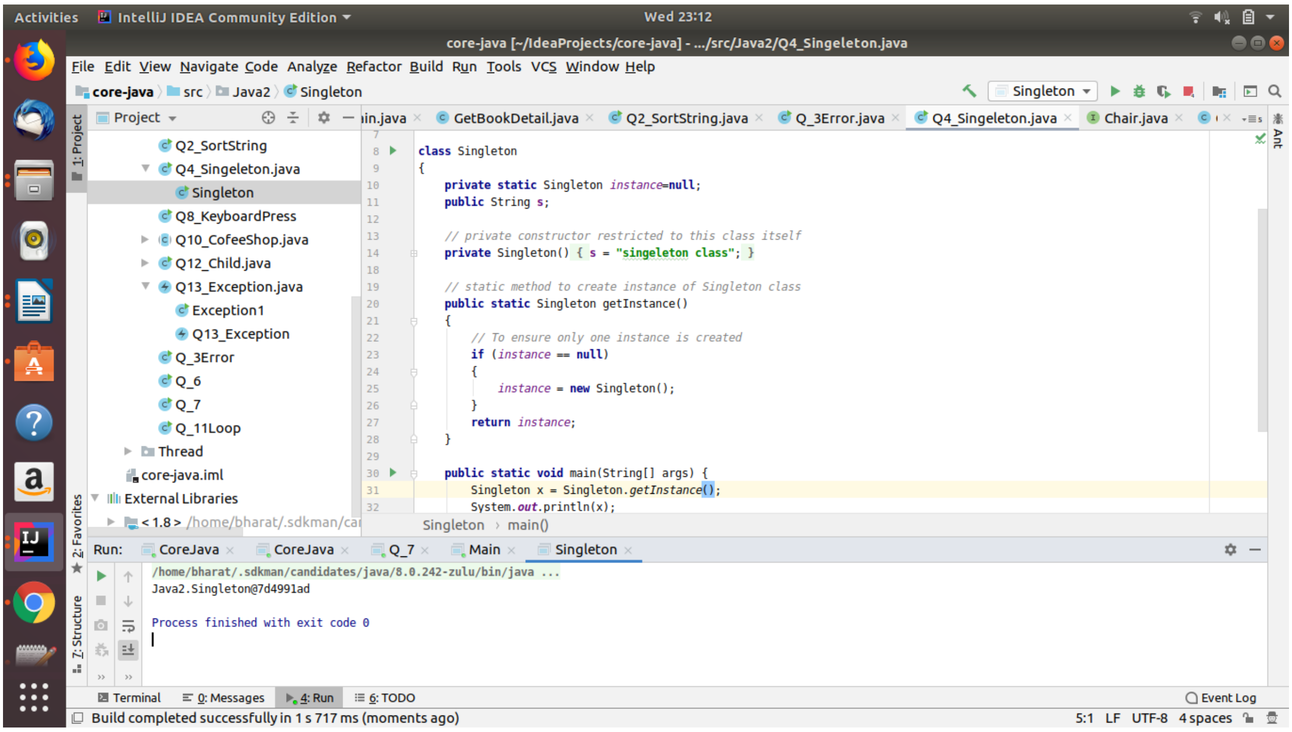
```
package Java2;/*

Created by
bharat
on 18/02/20
*/

public class Q_3Error {
    public static void main(String[] args) {
        try{
            Class.forName("Hello World");

        }
        catch (ClassNotFoundException e)
        {
          e.printStackTrace();
        }
```

```
/home/bharat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
java.lang.ClassNotFoundException: Hello World
	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)
	at java.lang.Class.forName0(Native Method)
	at java.lang.Class.forName(Class.java:264)
	at Java2.Q_3Error.main(Q_3Error.java:11)

Process finished with exit code 0
```

Build completed successfully in 1 s 771 ms (moments ago)

Q.4 WAP to create singleton class.

Code reference(Q4_Singeleton.java)



core-java [~/IdeaProjects/core-java] - .../src/Java2/Q4_Singeleton.java

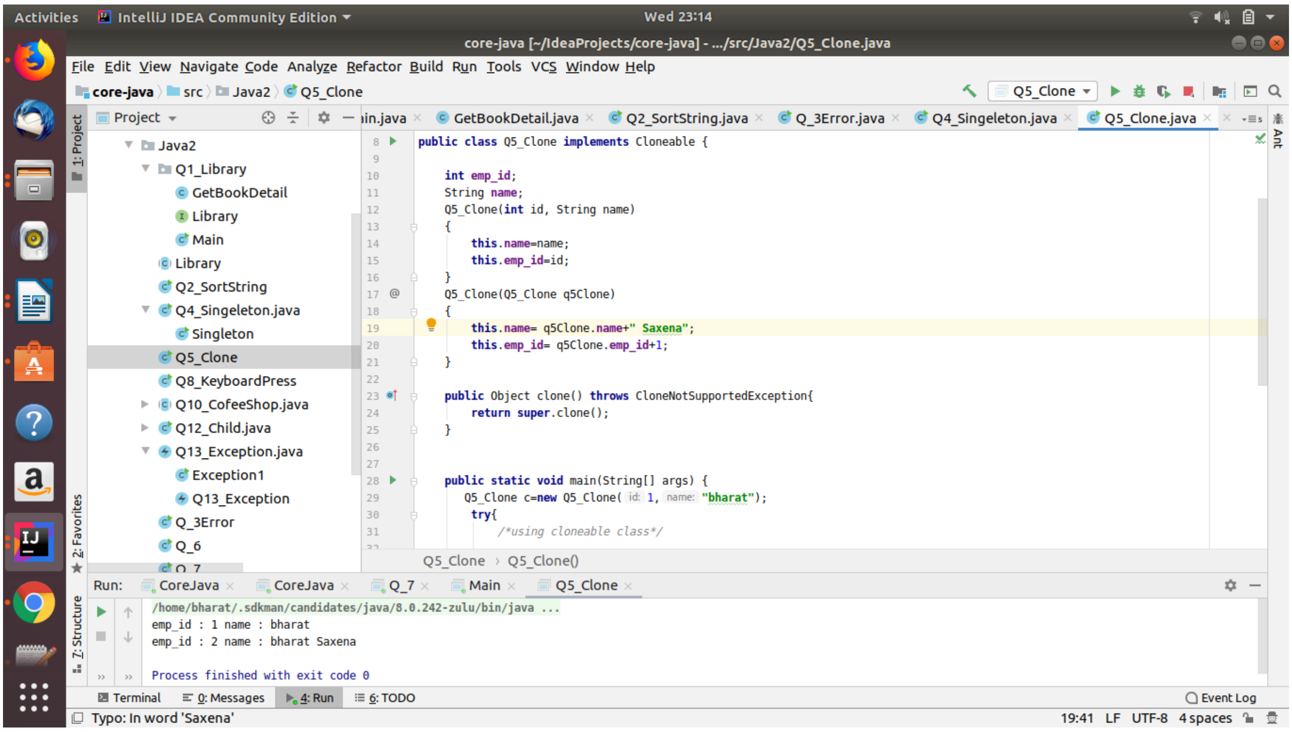
```java
class Singleton
{
    private static Singleton instance=null;
    public String s;

    // private constructor restricted to this class itself
    private Singleton() { s = "singeleton class"; }

    // static method to create instance of Singleton class
    public static Singleton getInstance()
    {
        // To ensure only one instance is created
        if (instance == null)
        {
            instance = new Singleton();
        }
        return instance;
    }

    public static void main(String[] args) {
        Singleton x = Singleton.getInstance();
        System.out.println(x);
```

```
/home/bharat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Java2.Singleton@7d4991ad

Process finished with exit code 0
```

Build completed successfully in 1 s 717 ms (moments ago)

Q.5 WAP to show object cloning in java using cloneable and copy constructor both.

Code reference(Q_5 clone.java);



core-java [~/IdeaProjects/core-java] - .../src/Java2/Q5_Clone.java

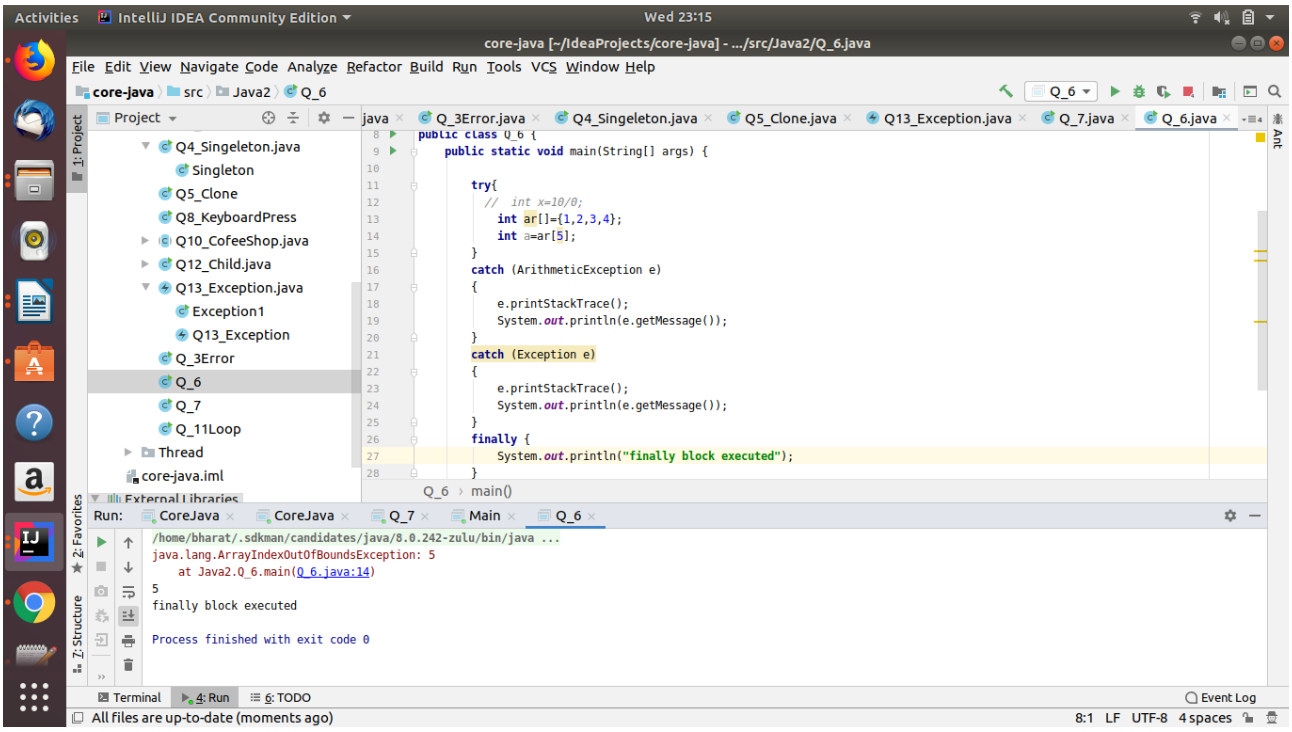
```java
public class Q5_Clone implements Cloneable {

    int emp_id;
    String name;
    Q5_Clone(int id, String name)
    {
        this.name=name;
        this.emp_id=id;
    }
    Q5_Clone(Q5_Clone q5Clone)
    {
        this.name= q5Clone.name+" Saxena";
        this.emp_id= q5Clone.emp_id+1;
    }

    public Object clone() throws CloneNotSupportedException{
        return super.clone();
    }


    public static void main(String[] args) {
       Q5_Clone c=new Q5_Clone( id: 1, name: "bharat");
        try{
            /*using cloneable class*/
```

```
/home/bharat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
emp_id : 1 name : bharat
emp_id : 2 name : bharat Saxena

Process finished with exit code 0
```

Typo: In word 'Saxena'

Q.6 WAP showing try, multi-catch and finally blocks

code reference(Q-6.java)

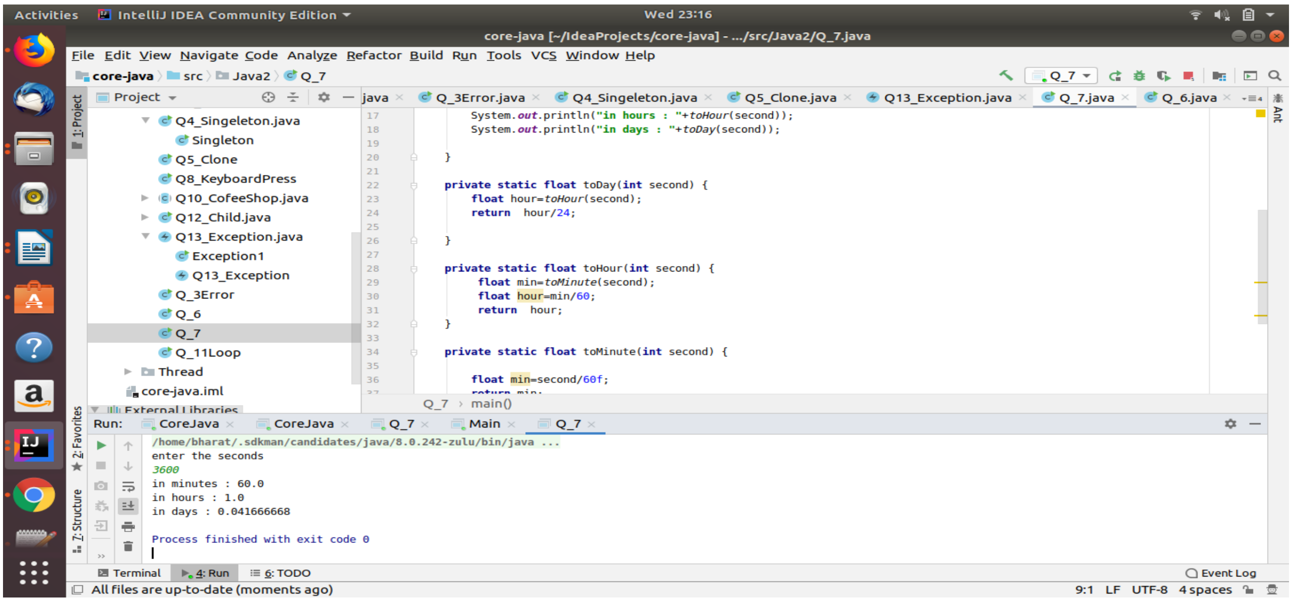


```java
public class Q_6 {
    public static void main(String[] args) {

        try{
        //  int x=10/0;
            int ar[]={1,2,3,4};
            int a=ar[5];
        }
        catch (ArithmeticException e)
        {
            e.printStackTrace();
            System.out.println(e.getMessage());
        }
        catch (Exception e)
        {
            e.printStackTrace();
            System.out.println(e.getMessage());
        }
        finally {
            System.out.println("finally block executed");
        }
```

```
/home/bharat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
java.lang.ArrayIndexOutOfBoundsException: 5
	at Java2.Q_6.main(Q_6.java:14)
5
finally block executed

Process finished with exit code 0
```

Q.7 WAP to convert seconds into days, hours, minutes and seconds.

Code reference(Q7.java)

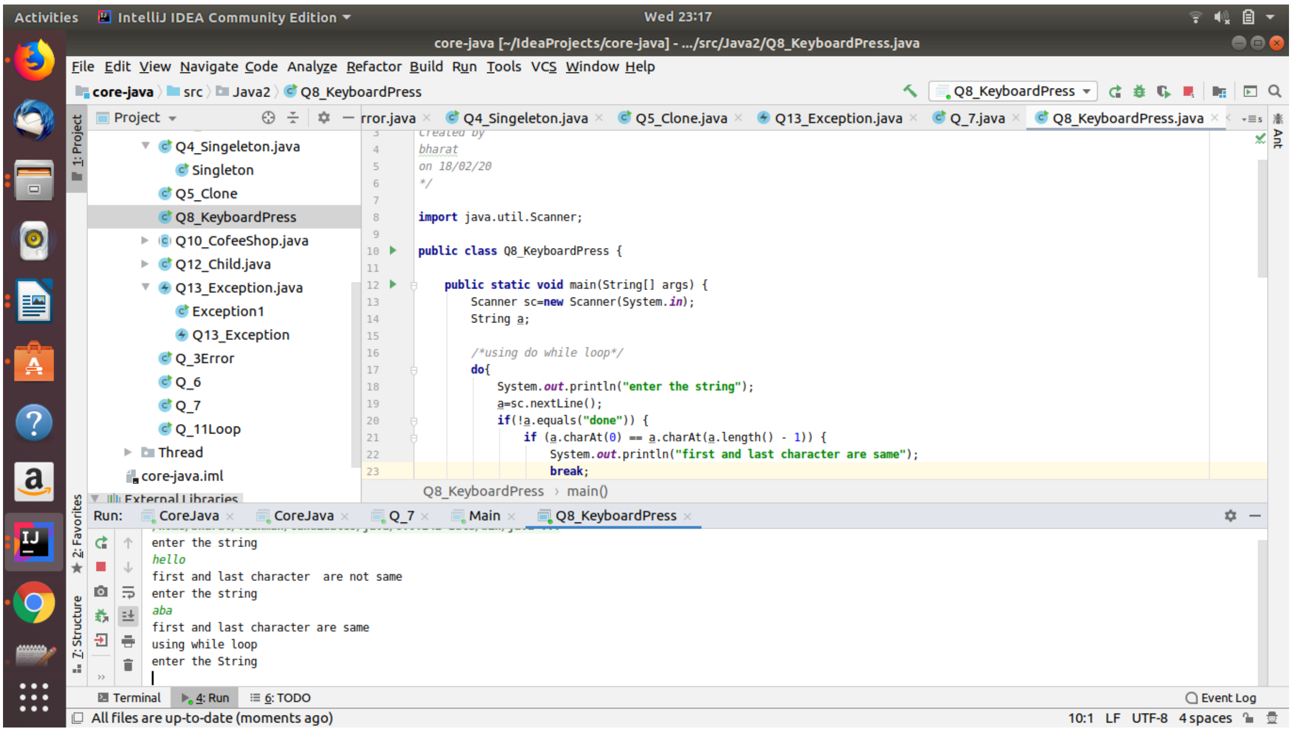


```java
        System.out.println("in hours : "+toHour(second));
        System.out.println("in days : "+toDay(second));

    }

    private static float toDay(int second) {
        float hour=toHour(second);
        return  hour/24;

    }

    private static float toHour(int second) {
         float min=toMinute(second);
         float hour=min/60;
         return  hour;
    }

    private static float toMinute(int second) {

        float min=second/60f;
        return min;
```

```
/home/bharat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
enter the seconds
3600
in minutes : 60.0
in hours : 1.0
in days : 0.041666668

Process finished with exit code 0
```

Q.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
a)while statement
b)do-while statement

code reference(Q8_KeyboardPress.java)



core-java [~/IdeaProjects/core-java] - .../src/Java2/Q8_KeyboardPress.java

```
Created by
bharat
on 18/02/20
*/

import java.util.Scanner;

public class Q8_KeyboardPress {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        String a;

        /*using do while loop*/
        do{
            System.out.println("enter the string");
            a=sc.nextLine();
            if(!a.equals("done")) {
                if (a.charAt(0) == a.charAt(a.length() - 1)) {
                    System.out.println("first and last character are same");
                    break;
```

```
enter the string
hello
first and last character  are not same
enter the string
aba
first and last character are same
using while loop
enter the String
```

All files are up-to-date (moments ago)

Q.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.

Ans.

// Chair class interface

```
public interface Chair {
     public String getPrice();
     public String getChairType();
     public String getStressResult();
     public String fireTest();
}
```

```
// for implemenmting the chair interface methods
package Furniture;/*
Created by
bharat
on 19/02/20
*/
public class ChairClass {
}
```

```
//for wooden chair
class WoodenChair implements Chair
{
    @Override
    public String getPrice() {
        return null;
    }
    @Override
    public String getChairType() {
        return null;
    }
    @Override
    public String getStressResult() {
        return null;
    }
    @Override
    public String fireTest() {
        return null;
    }
}

//for metal chair
class MetalChair implements Chair
{
    @Override
    public String getPrice() {
        return null;
    }
    @Override
    public String getChairType() {
        return null;
    }
    @Override
    public String getStressResult() {
        return null;
    }
    @Override
    public String fireTest() {
        return null;
    }
}



// table class interface

package Furniture;/*

Created by
bharat
on 19/02/20
*/
public interface Table {
    public String getPrice();
    public String getChairType();
    public String getStressResult();
    public String fireTest();
}



// implemnting all methods od table interface

package Furniture;/*
```

```java
Created by
bharat
on 19/02/20
*/
public class TableClass {
}

//for wooden table
class WoodenTable implements Table
{
    @Override
    public String getPrice() {
        return null;
    }
    @Override
    public String getChairType() {
        return null;
    }
    @Override
    public String getStressResult() {
        return null;
    }
    @Override
    public String fireTest() {
        return null;
    }
}

//for metal table
class MetalTable implements Table
{
    @Override
    public String getPrice() {
        return null;
    }
    @Override
    public String getChairType() {
        return null;
    }
    @Override
    public String getStressResult() {
        return null;
    }
    @Override
    public String fireTest() {
        return null;
    }
}
```

Q.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

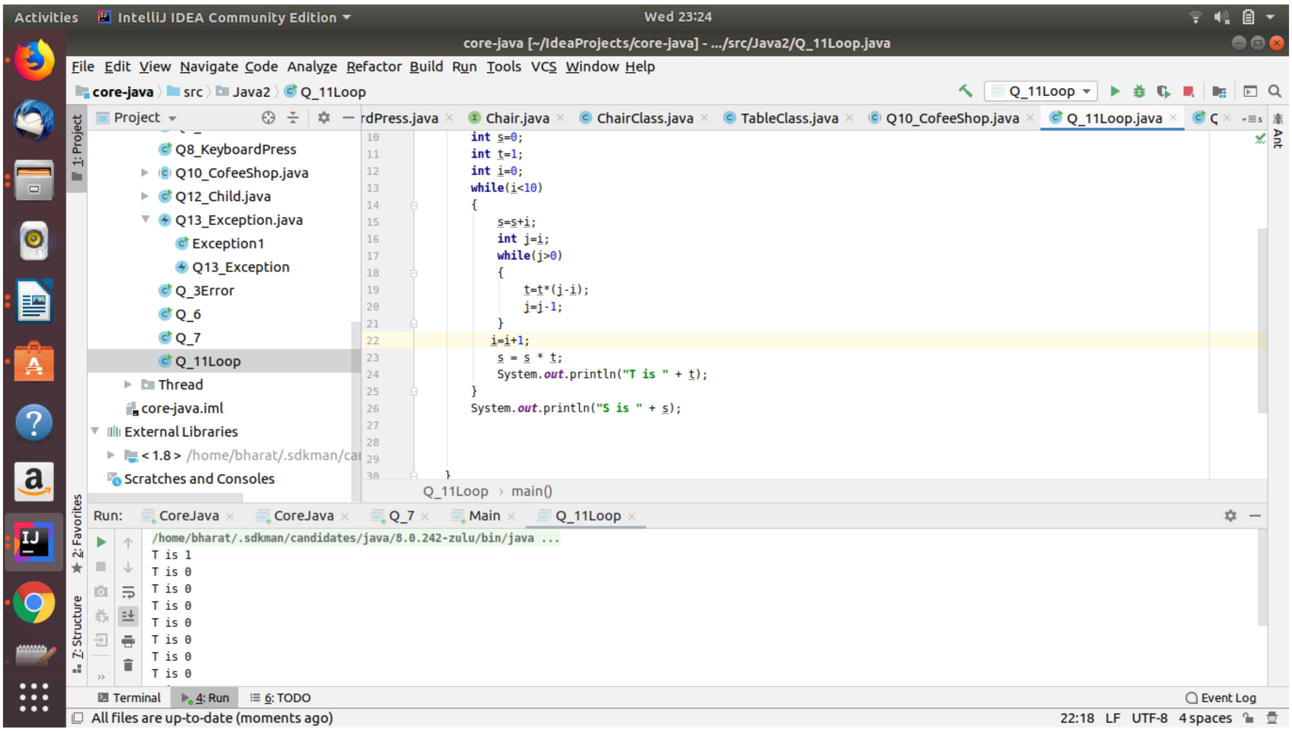
Ans..

```java
package Java2;/*
Created by
bharat
on 19/02/20
*/
abstract public class Q10_CofeeShop extends Barista {

}
abstract class Customer{
    boolean coffeePrepared=false;
    abstract void payment();
    abstract int getToken();
    abstract boolean isCoffeePrepared();
}
abstract class Cashier extends Customer{
    abstract void getOrder();
    abstract void createOrder();
    abstract int waitCustomer();
    void getPayment(){
        createOrder();
        waitCustomer();
    }
}
abstract class Barista extends Cashier{
    boolean coffeePrepared;
    abstract void getNextOrder();
    boolean prepareCoffee(){
         System.out.println("processing coffee");
         coffeePrepared=true;
         return coffeePrepared;
    }
    abstract int completedOrder();
    String orderCompletedNotification(){
         return "order completed";
    }
}
```

Q.11Convert 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);
```

code reference (Q_11.java)



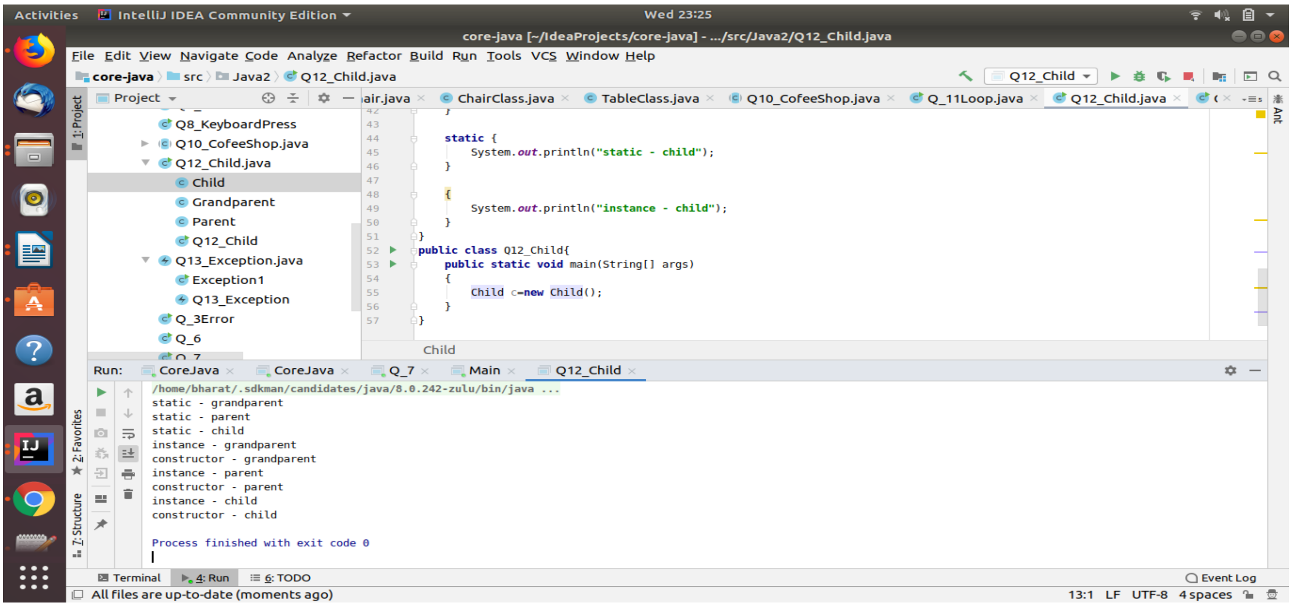
```
        int s=0;
        int t=1;
        int i=0;
        while(i<10)
        {
            s=s+i;
            int j=i;
            while(j>0)
            {
                t=t*(j-i);
                j=j-1;
            }
           i=i+1;
            s = s * t;
            System.out.println("T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



    }
```

```
/home/bharat/.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
```

Q.12 What will be the output on new Child(); ?

```
  class Parent extends Grandparent {
      {
      System.out.println("instance - parent");
      }

  public Parent() {
  System.out.println("
  }

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

class Grandparent {

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

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

  public Grandparent() {
  System.out.println("
  }
}

class Child extends Parent {

  public Child() {
  System.out.println("
  }

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

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

Ans. Code reference (Q12_Child.java)

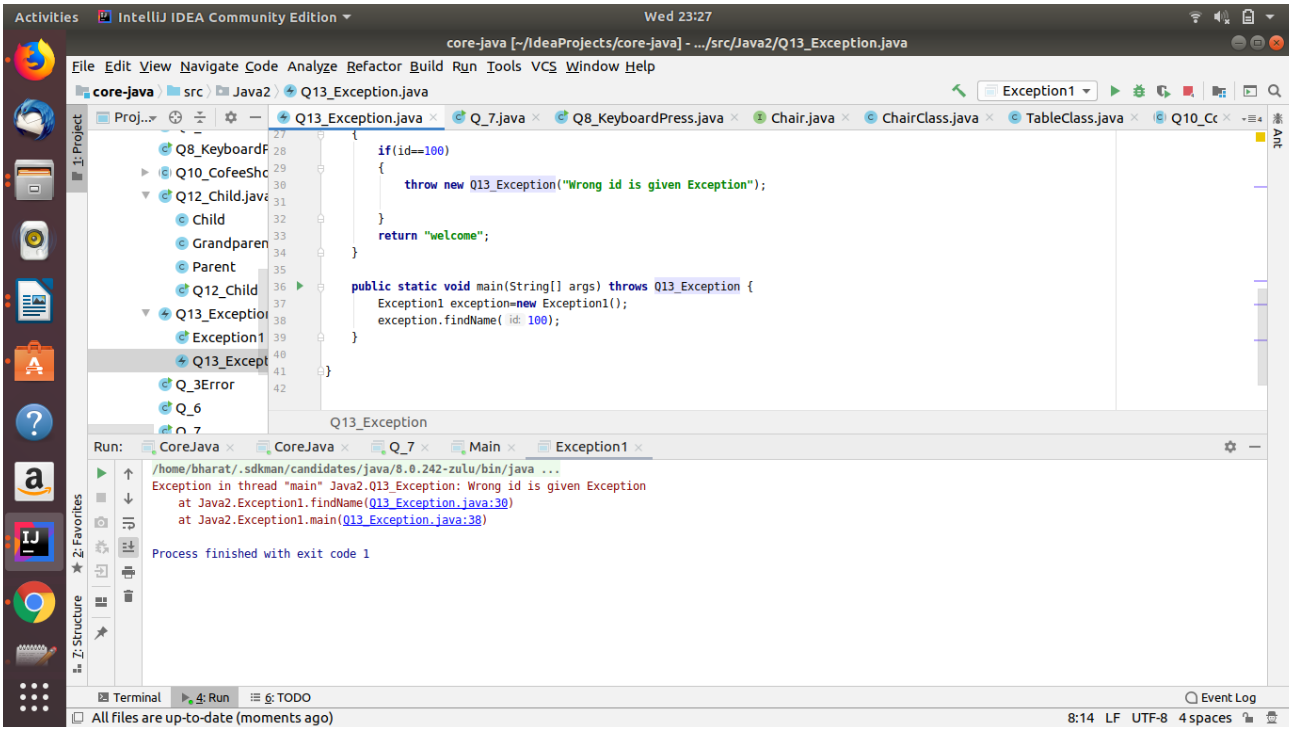


```
    }

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

    {
        System.out.println("instance - child");
    }
}
public class Q12_Child{
    public static void main(String[] args)
    {
        Child c=new Child();
    }
}
```

```
/home/bharat/.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
```

Q.13 Create a custom exception that do not have any stack trace.

Code reference(Q13_Exception.java)



```
    {
        if(id==100)
        {
            throw new Q13_Exception("Wrong id is given Exception");

        }
        return "welcome";
    }

    public static void main(String[] args) throws Q13_Exception {
        Exception1 exception=new Exception1();
        exception.findName( id: 100);
    }

}
```

```
/home/bharat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Exception in thread "main" Java2.Q13_Exception: Wrong id is given Exception
    at Java2.Exception1.findName(Q13_Exception.java:30)
    at Java2.Exception1.main(Q13_Exception.java:38)

Process finished with exit code 1
```