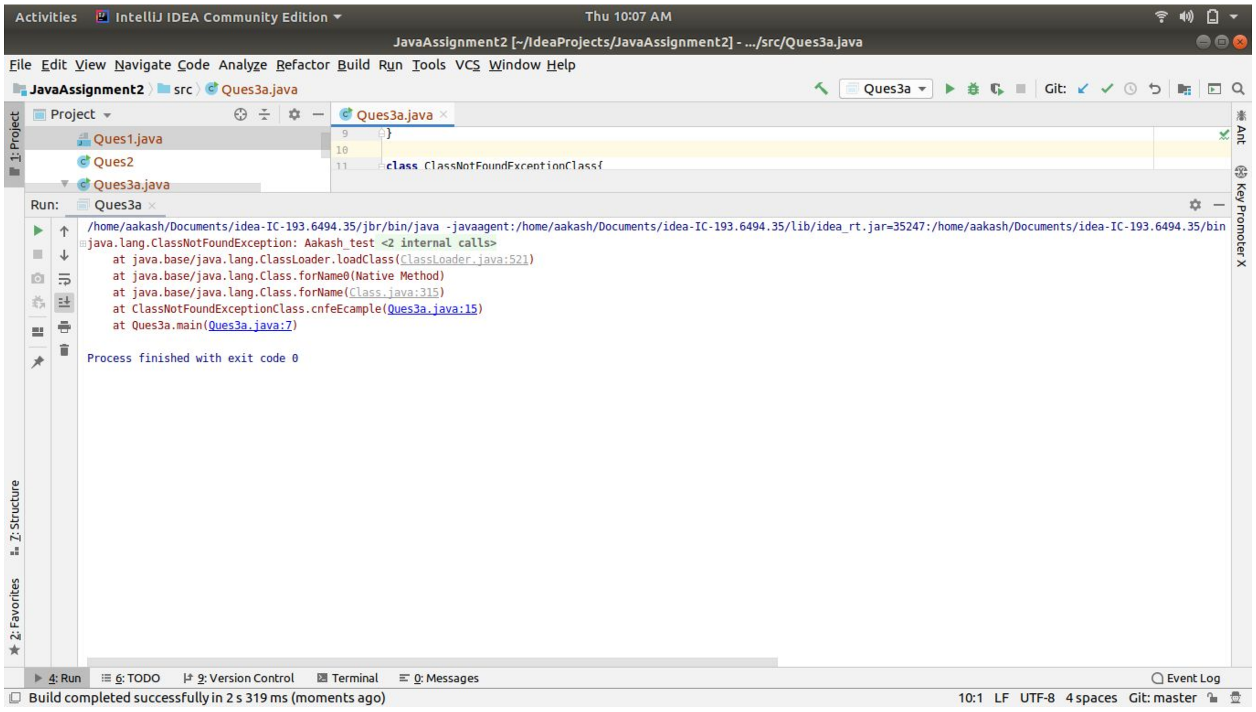
```java
//Ques 1: Create Java classes having suitable attributes for Library management system.Use OOPs concepts in your design.
// Also try to use interfaces and abstract classes.

enum BookFormat {
  HARDCOVER,
  PAPERBACK,
  AUDIO_BOOK,
  EBOOK,
  NEWSPAPER,
  MAGAZINE,
  JOURNAL
}
enum BookStatus {
  AVAILABLE,
  RESERVED,
  LOANED,
  LOST
}
enum ReservationStatus{
  WAITING,
  PENDING,
  CANCELED,
  NONE
}
enum AccountStatus{
  ACTIVE,
  CLOSED,
  CANCELED,
  BLACKLISTED,
  NONE
}
class Address {
  private String streetAddress;
  private String city;
  private String state;
  private String zipCode;
  private String country;
}
class Person {
  private String name;
  private Address address;
  private String email;
  private String phone;
}
class Constants {
  public static final int MAX_BOOKS_ISSUED_TO_A_USER = 5;
  public static final int MAX_LENDING_DAYS = 10;
}
```

```
// For simplicity, we are not defining getter and setter functions. The reader can
// assume that all class attributes are private and accessed through their respective
// public getter methods and modified only through their public methods function.
public abstract class Account {
  private String id;
  private String password;
  private AccountStatus status;
  private Person person;
  public boolean resetPassword();
}
public class Librarian extends Account {
  public boolean addBookItem(BookItem bookItem);
  public boolean blockMember(Member member);
  public boolean unBlockMember(Member member);
}
public class Member extends Account {
  private Date dateOfMembership;
  private int totalBooksCheckedout;
  public int getTotalBooksCheckedout();
  public boolean reserveBookItem(BookItem bookItem);
  private void incrementTotalBooksCheckedout();
  public boolean checkoutBookItem(BookItem bookItem) {
  }
  private void checkForFine(String bookItemBarcode) {
  }
  public void returnBookItem(BookItem bookItem) {
  }
  public boolean renewBookItem(BookItem bookItem) {
  }
}
public class BookReservation {
  private Date creationDate;
  private ReservationStatus status;
  private String bookItemBarcode;
  private String memberId;
  public static BookReservation fetchReservationDetails(String barcode);
}
public class BookLending {
  private Date creationDate;
  private Date dueDate;
  private Date returnDate;
  private String bookItemBarcode;
  private String memberId;
  public static void lendBook(String barcode, String memberId);
  public static BookLending fetchLendingDetails(String barcode);
}
class Fine {
  private Date creationDate;
  private double bookItemBarcode;
```

```
    private String memberId;
    public static void collectFine(String memberId, long days) {}
}
```

```
//Ques 2: WAP for sorting string without using string Methods?.

import java.util.Scanner;

public class Ques2 {
  public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);
     System.out.println("Please enter a String: ");
     String original = sc.nextLine();
     int j = 0;
     char temp = 0;

     char[] chars = original.toLowerCase().toCharArray();

     for (int i = 1; i < chars.length; i++) {

        for (j = 0; j < chars.length; j++) {

           if (chars[j] > chars[i]) {
              temp = chars[i];
              chars[i] = chars[j];
              chars[j] = temp;
           }

        }

     }

     for (int k = 0; k < chars.length; k++) {
        if(chars[k]==' '){
           continue;
        }
        System.out.print(chars[k]);
        System.out.print(" ");
     }

  }

}
```

```
//Ques 3:a:  WAP to produce ClassNotFoundException exception.

public class Ques3a {
  public static void main(String[] args) {

  ClassNotFoundExceptionClass cnfe = new ClassNotFoundExceptionClass();
  cnfe.cnfeEcample();
  }
}

class ClassNotFoundExceptionClass{
  public void cnfeEcample(){
     try
     {
        Class.forName("Aakash_test");
     }
     catch (ClassNotFoundException ex)
     {
        ex.printStackTrace();
     }
  }
}
```

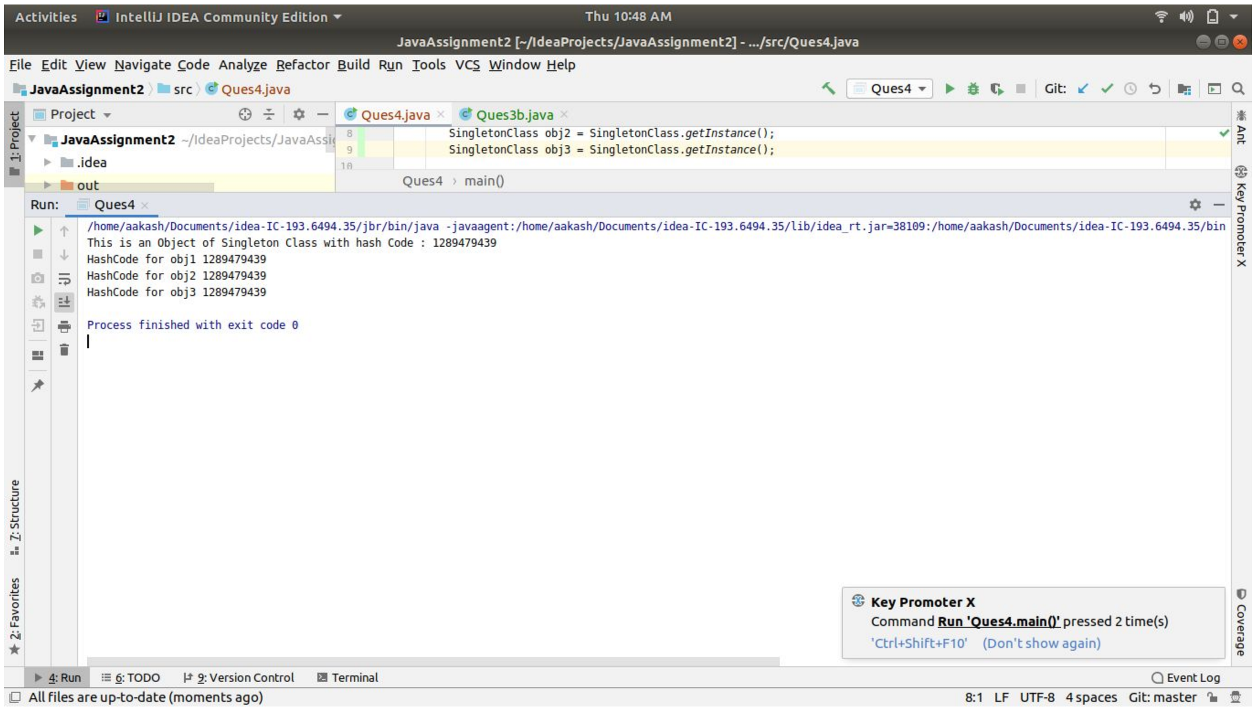


```
/home/aakash/Documents/idea-IC-193.6494.35/jbr/bin/java -javaagent:/home/aakash/Documents/idea-IC-193.6494.35/lib/idea_rt.jar=35247:/home/aakash/Documents/idea-IC-193.6494.35/bin
java.lang.ClassNotFoundException: Aakash_test <2 internal calls>
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:521)
    at java.base/java.lang.Class.forName0(Native Method)
    at java.base/java.lang.Class.forName(Class.java:315)
    at ClassNotFoundExceptionClass.cnfeEcample(Ques3a.java:15)
    at Ques3a.main(Ques3a.java:7)

Process finished with exit code 0
```

Build completed successfully in 2 s 319 ms (moments ago)

```java
//Ques 3 b: WAP to produce NoClassDefFoundError exception.


public class Ques3b {

   public static void main(String[] args) {

       noclassdeffounderr g = new noclassdeffounderr();
       g.greet();

   }
}

class noclassdeffounderr
{
   void greet()
   {
       System.out.println("hello!");
   }
}
```

```java
//Ques 4: WAP to create singleton class.

public class Ques4 {

  public static void main(String[] args) {

      SingletonClass obj1 = SingletonClass.getInstance();
      SingletonClass obj2 = SingletonClass.getInstance();
      SingletonClass obj3 = SingletonClass.getInstance();

//
      System.out.println("HashCode for obj1 "+ obj1.hashCode());
      System.out.println("HashCode for obj2 "+ obj2.hashCode());
      System.out.println("HashCode for obj3 "+ obj3.hashCode());
  }
}

class SingletonClass{

  private static SingletonClass singletonObject= null;

  private SingletonClass(){
      System.out.print("This is an Object of Singleton Class with hash Code : ");
      System.out.println(this.hashCode());
  }

  public static SingletonClass getInstance(){
      if(singletonObject==null){
          singletonObject = new SingletonClass();
      }
      return singletonObject;
  }
}
```



```
SingletonClass obj2 = SingletonClass.getInstance();
SingletonClass obj3 = SingletonClass.getInstance();
```

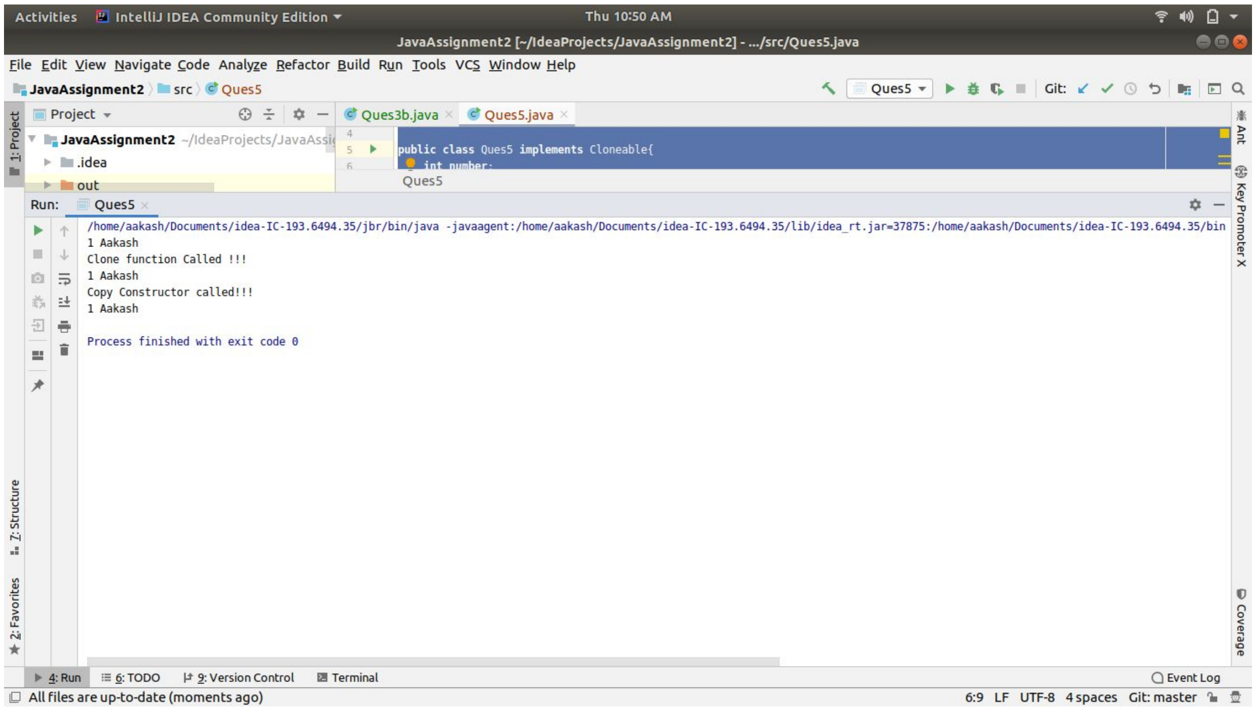
```
/home/aakash/Documents/idea-IC-193.6494.35/jbr/bin/java -javaagent:/home/aakash/Documents/idea-IC-193.6494.35/lib/idea_rt.jar=38109:/home/aakash/Documents/idea-IC-193.6494.35/bin
This is an Object of Singleton Class with hash Code : 1289479439
HashCode for obj1 1289479439
HashCode for obj2 1289479439
HashCode for obj3 1289479439

Process finished with exit code 0
```

Key Promoter X
Command Run 'Ques4.main()' pressed 2 time(s)
'Ctrl+Shift+F10' (Don't show again)

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

Activities
IntelliJ IDEA Community Edition
Thu 10:48 AM
JavaAssignment2 [~/IdeaProjects/JavaAssignment2] - .../src/Ques4.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
JavaAssignment2 src Ques4.java
Ques4
Git:
Project
JavaAssignment2 ~/IdeaProjects/JavaAssi
.idea
out
Ques4.java
Ques3b.java
Ques4 › main()
Run: Ques4
4: Run
6: TODO
9: Version Control
Terminal
Event Log
8:1 LF UTF-8 4 spaces Git: master

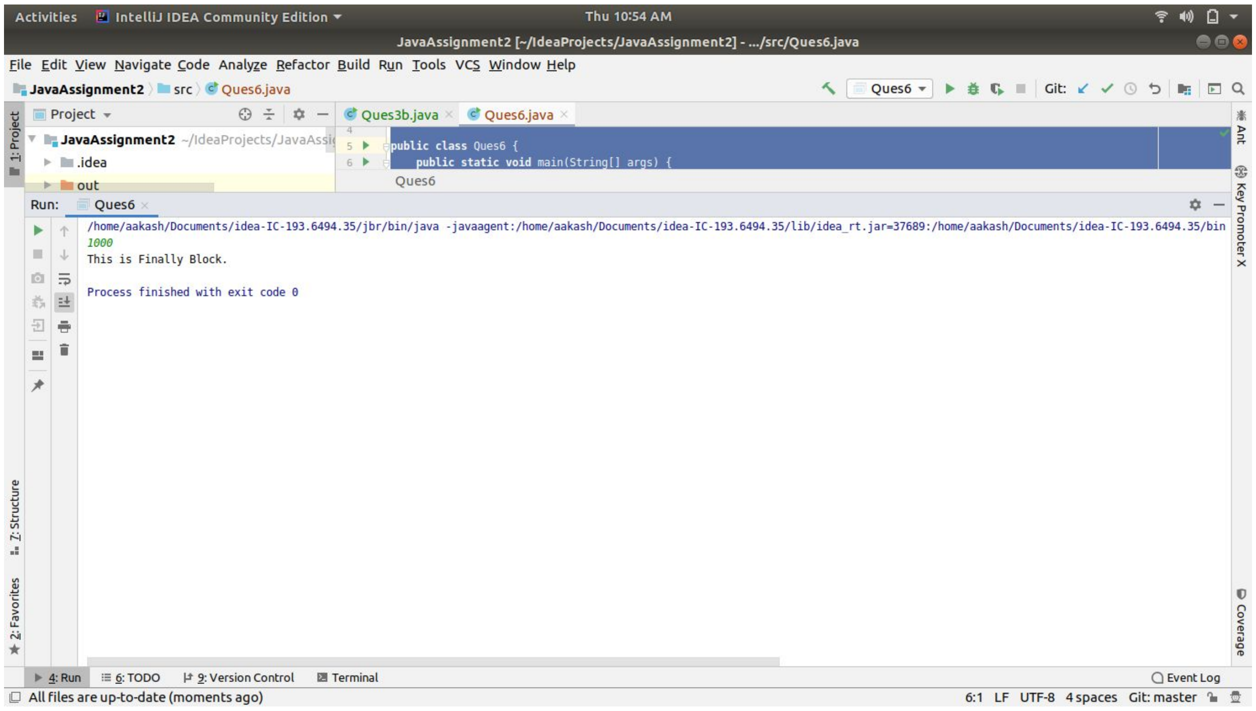
```java
//Ques 5: WAP to show object cloning in java using cloneable and copy constructor both.

import java.lang.Cloneable;

public class Ques5 implements Cloneable{
  int number;
  String str;

  Ques5(int number , String str){
     this.number = number;
     this.str= str;
  }

  Ques5(Ques5 obj){
     System.out.println("Copy Constructor called!!!");
     number= obj.number;
     str= obj.str;

  }

  public Object clone()throws CloneNotSupportedException{
     System.out.println("Clone function Called !!!");

     return super.clone();
  }

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


     Ques5 originalObject1 = new Ques5(1,"Aakash");
        System.out.println(originalObject1.number+" " + originalObject1.str);

     Ques5 cloneableObject2 = (Ques5) originalObject1.clone();
        System.out.println(cloneableObject2.number+" " + cloneableObject2.str);

     Ques5 copyConstructorObject3 =new Ques5(originalObject1);
        System.out.println(copyConstructorObject3.number+" " + copyConstructorObject3.str);

     }
     catch (CloneNotSupportedException cnse){}


  }
}
```


Activities
IntelliJ IDEA Community Edition
Thu 10:50 AM
JavaAssignment2 [~/IdeaProjects/JavaAssignment2] - .../src/Ques5.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
JavaAssignment2 src Ques5
Ques3b.java
Ques5.java
Project
JavaAssignment2 ~/IdeaProjects/JavaAssig
.idea
out
Run: Ques5
4: Run
6: TODO
9: Version Control
Terminal
Event Log
All files are up-to-date (moments ago)
6:9 LF UTF-8 4 spaces Git: master

```java
public class Ques5 implements Cloneable{
    int number;
```

```
/home/aakash/Documents/idea-IC-193.6494.35/jbr/bin/java -javaagent:/home/aakash/Documents/idea-IC-193.6494.35/lib/idea_rt.jar=37875:/home/aakash/Documents/idea-IC-193.6494.35/bin
1 Aakash
Clone function Called !!!
1 Aakash
Copy Constructor called!!!
1 Aakash

Process finished with exit code 0
```

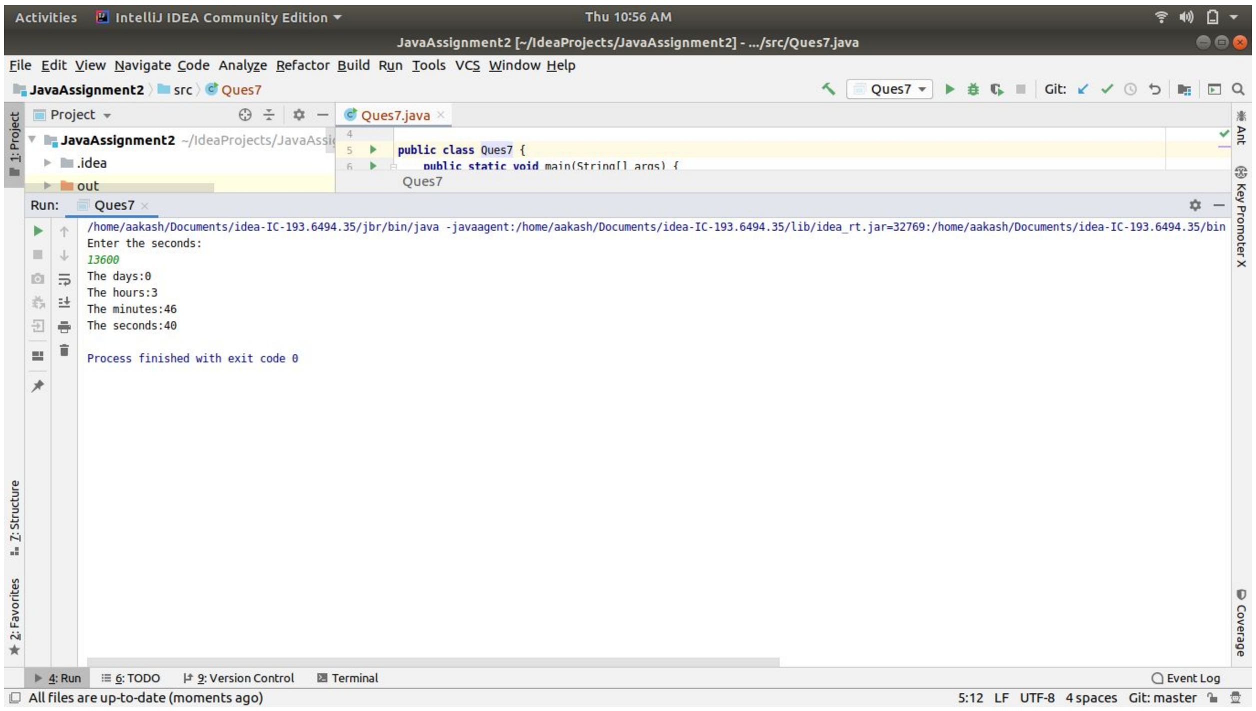
```java
//Ques 6: WAP showing try, multi-catch and finally blocks.

import java.util.Scanner;

public class Ques6 {
  public static void main(String[] args) {

      TryCatch obj1 = new TryCatch();
      obj1.tryCatch();

  }

}

class TryCatch
{
  public void tryCatch()
  {
      Scanner scn = new Scanner(System.in);
      try
      {
          int n = Integer.parseInt(scn.nextLine());
          if (99%n == 0)
              System.out.println(n + " is a factor of 99");
      }
      catch (ArithmeticException ex)
      {
          System.out.println("Arithmetic " + ex);
      }
      catch (NumberFormatException ex)
      {
          System.out.println("Number Format Exception " + ex);
      }
      finally {
          System.out.println("This is Finally Block.");
      }
  }
}
```



```
public class Ques6 {
    public static void main(String[] args) {
```

```
/home/aakash/Documents/idea-IC-193.6494.35/jbr/bin/java -javaagent:/home/aakash/Documents/idea-IC-193.6494.35/lib/idea_rt.jar=37689:/home/aakash/Documents/idea-IC-193.6494.35/bin
1000
This is Finally Block.

Process finished with exit code 0
```

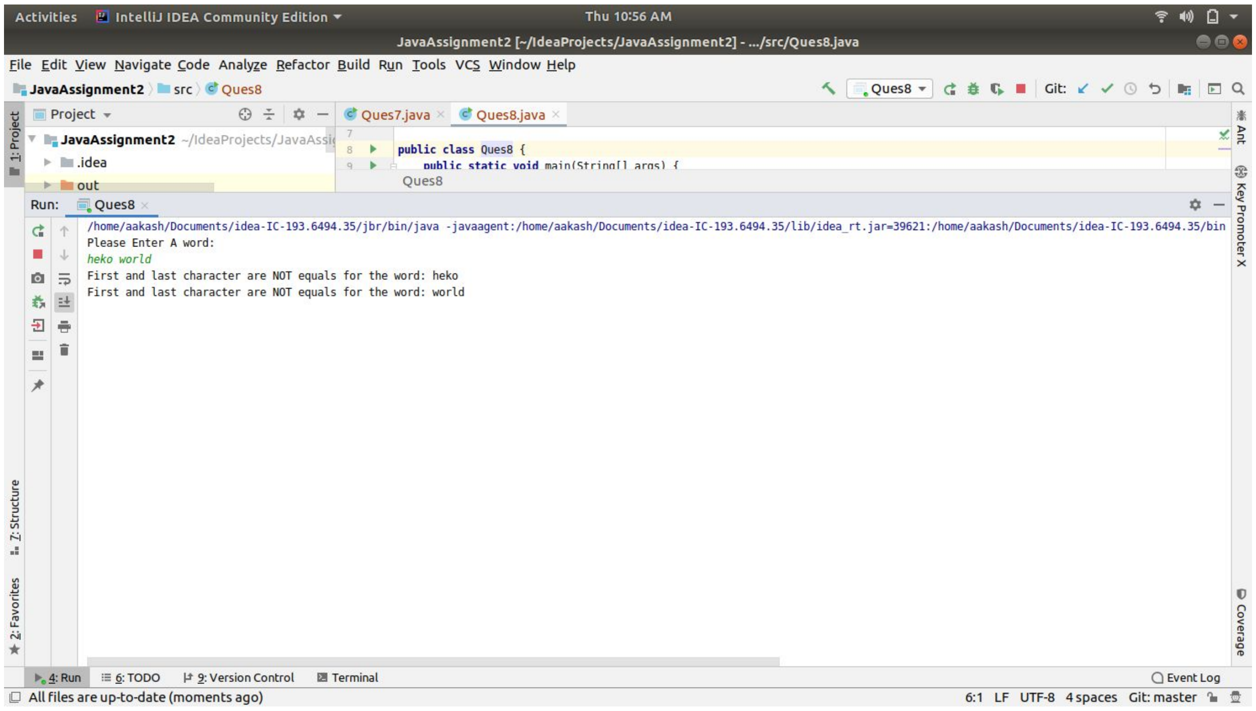
```java
//Ques 7: WAP to convert seconds into days, hours, minutes and seconds.

import java.util.Scanner;

public class Ques7 {
  public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the seconds: ");
      int n = sc.nextInt();

      int days = n / (24*3600);
      n = n % (24*3600);
      int hour = n / 3600;
      n = n % 3600;
      int min = n / 60;
      n = n % 60;

      System.out.println("The days:"+days);
      System.out.println("The hours:"+hour);
      System.out.println("The minutes:"+min);
      System.out.println("The seconds:"+n);
  }
}
```



```java
public class Ques7 {
    public static void main(String[] args) {
```

```
/home/aakash/Documents/idea-IC-193.6494.35/jbr/bin/java -javaagent:/home/aakash/Documents/idea-IC-193.6494.35/lib/idea_rt.jar=32769:/home/aakash/Documents/idea-IC-193.6494.35/bin
Enter the seconds: 
13600
The days:0
The hours:3
The minutes:46
The seconds:40

Process finished with exit code 0
```

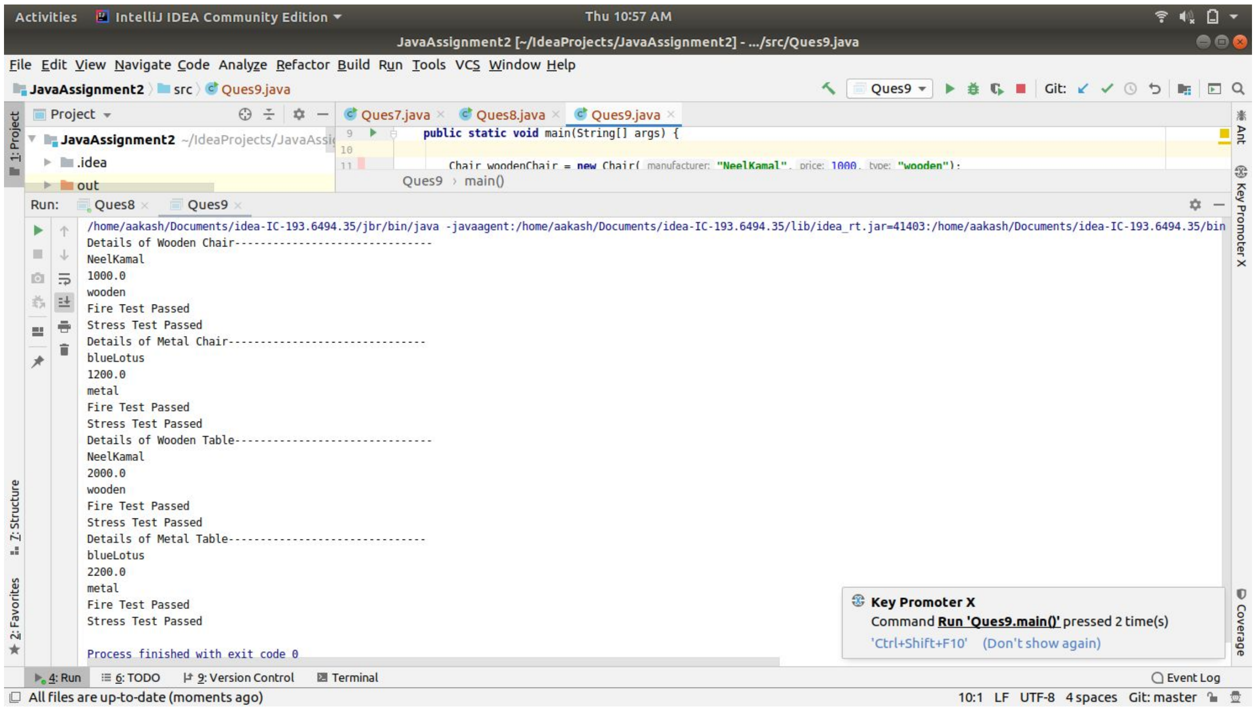
```
//Ques 8: WAP to read words from the keyboard until the word done is entered. For each word except done,
// report whether its first character is equal to  its last character.
// For the required loop, use a
//        a)while statement

import java.util.Scanner;

public class Ques8 {
  public static void main(String[] args) {
      Scanner keyboard = new Scanner(System.in);
      System.out.println("Please Enter A word: ");
      String word = keyboard.next();
      while(!word.equals("done"))
      {
          if(word.charAt(0) == word.charAt(word.length() - 1))
              System.out.println("First and last character are equals for the word: " + word);

          else
              System.out.println("First and last character are NOT equals for the word: " + word);

          word = keyboard.next();
      }
  }
}
```


Activities
IntelliJ IDEA Community Edition
Thu 10:56 AM
JavaAssignment2 [~/IdeaProjects/JavaAssignment2] - .../src/Ques8.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
JavaAssignment2 src Ques8
Ques8
Git:
Project
JavaAssignment2 ~/IdeaProjects/JavaAssi
.idea
out
Ques7.java
Ques8.java
Run: Ques8
4: Run
6: TODO
9: Version Control
Terminal
Event Log
All files are up-to-date (moments ago)
6:1 LF UTF-8 4 spaces Git: master

```
public class Ques8 {
    public static void main(String[] args) {
```

```
/home/aakash/Documents/idea-IC-193.6494.35/jbr/bin/java -javaagent:/home/aakash/Documents/idea-IC-193.6494.35/lib/idea_rt.jar=39621:/home/aakash/Documents/idea-IC-193.6494.35/bin
Please Enter A word:
heko world
First and last character are NOT equals for the word: heko
First and last character are NOT equals for the word: world
```

```java
//Ques 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.

import javafx.scene.control.Tab;

import java.util.Date;

public class Ques9 {
  public static void main(String[] args) {

      Chair woodenChair = new Chair("NeelKamal",1000,"wooden");
      Chair metalChair = new Chair("blueLotus",1200,"metal");

      Table woodenTable = new Table("NeelKamal",2000,"wooden");
      Table metalTable = new Table("blueLotus",2200,"metal");

      System.out.println("Details of Wooden Chair-------------------------------");
      System.out.println(woodenChair.manufacturer);
      System.out.println(woodenChair.price);
      System.out.println(woodenChair.type);
      System.out.println(woodenChair.fireTest());
      System.out.println(woodenChair.stessTest());

      System.out.println("Details of Metal Chair-------------------------------");
      System.out.println(metalChair.manufacturer);
      System.out.println(metalChair.price);
      System.out.println(metalChair.type);
      System.out.println(metalChair.fireTest());
      System.out.println(metalChair.stessTest());

      System.out.println("Details of Wooden Table-------------------------------");
      System.out.println(woodenTable.manufacturer);
      System.out.println(woodenTable.price);
      System.out.println(woodenTable.type);
      System.out.println(woodenTable.fireTest());
      System.out.println(woodenTable.stessTest());

      System.out.println("Details of Metal Table-------------------------------");
      System.out.println(metalTable.manufacturer);
      System.out.println(metalTable.price);
      System.out.println(metalTable.type);
      System.out.println(metalTable.fireTest());
      System.out.println(metalTable.stessTest());
  }

}

class Furniture{
  String manufacturer;
//    Date manufaturingDate;
  float price;

//    Furniture(){}
  Furniture(String manufacturer, float price){
      this.manufacturer = manufacturer;
//        this.manufaturingDate= manufaturingDate;
      this.price = price;
  }

  public String stessTest(){

      return "Stress Test Passed";
  }
  public String fireTest(){

      return "Fire Test Passed";
  }
}

class Chair extends Furniture{
  String type ;

  Chair(String manufacturer, float price, String type){
      super(manufacturer,price);
      this.type = type;
  }
}

class Table extends Furniture{
  String type ;

  Table(String manufacturer, float price, String type){
      super(manufacturer,price);
      this.type = type;
  }
}
```



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

        Chair woodenChair = new Chair( manufacturer: "NeelKamal", price: 1000, type: "wooden");
```

```
/home/aakash/Documents/idea-IC-193.6494.35/jbr/bin/java -javaagent:/home/aakash/Documents/idea-IC-193.6494.35/lib/idea_rt.jar=41403:/home/aakash/Documents/idea-IC-193.6494.35/bin
Details of Wooden Chair-----------------------------
NeelKamal
1000.0
wooden
Fire Test Passed
Stress Test Passed
Details of Metal Chair-----------------------------
blueLotus
1200.0
metal
Fire Test Passed
Stress Test Passed
Details of Wooden Table-----------------------------
NeelKamal
2000.0
wooden
Fire Test Passed
Stress Test Passed
Details of Metal Table-----------------------------
blueLotus
2200.0
metal
Fire Test Passed
Stress Test Passed

Process finished with exit code 0
```

Key Promoter X
Command Run 'Ques9.main()' pressed 2 time(s)
'Ctrl+Shift+F10' (Don't show again)

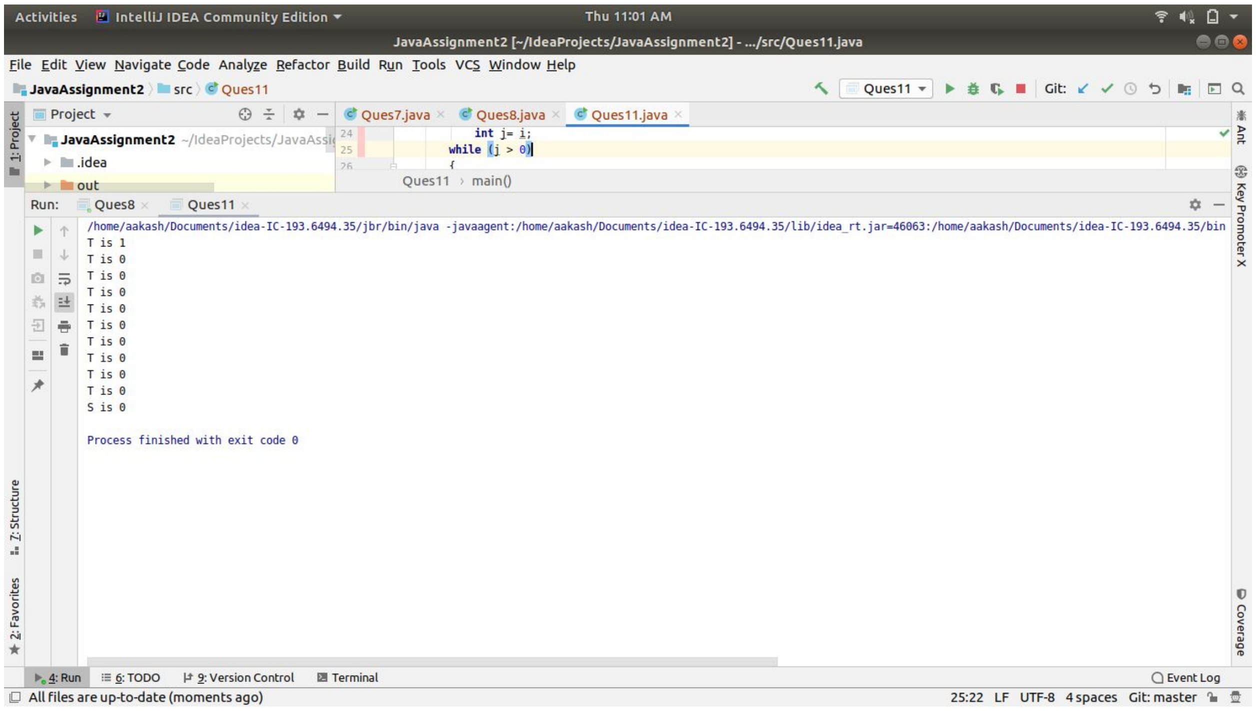
All files are up-to-date (moments ago)

```
// Qoes 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


public class Ques10 {
public static void main(String[] args) {}
  }


class Barista implements QueueOfPendingOrder,QueueOfCompletedOrder {
  String name;
  QueueOfPendingOrder queueOfPendingOrder;
  QueueOfCompletedOrder queueOfCompletedOrder;
  @Override
  public void remove(Order order) {
     queueOfPendingOrder.remove(order);
     System.out.println(order + "removed from Pending queue");
  }
  public void makeCoffee(Order order){
     System.out.println("Making coffee for " + order);
  }
  public void notifyAboutCompletedOrder(Customer customer,Order order) {
     System.out.println(customer + "your order " + order + " has been completed");
  }
  @Override
  public void addToCompleteOrderQueue(Order order) {
     queueOfCompletedOrder.addToCompleteOrderQueue(order);
     System.out.println(order + " added to Completed queue");
  }
```

```
}
class Cashier extends Order implements QueueOfPendingOrder {
  String name;
  QueueOfPendingOrder queueOfPendingOrder;
  List<Customer> customerList;
  public Cashier(String name,Long id) {
     super(id);
     this.name = name;
     customerList = new ArrayList<>();
  }
  String AcceptOrderAndAddCustomerToCustomerList(Customer customer,Order order,double cash) {
     customerList.add(customer);
     System.out.println("Accepted order");
     return "token";
  }
  void addOrderInOrderQueue(Order order){
     queueOfPendingOrder.add(order);
     System.out.println(order + " added to order queue");
  }
}
class Customer {
  private String name;
  private String token;
  Cashier cashier;
  Order order;
  double amount;
  void placeOrder() {
     token = cashier.AcceptOrderAndAddCustomerToCustomerList(this,order,amount);
     System.out.println("This is the order token: " + token);
  }
  boolean waitingState(){
     System.out.println("Customer" + this.name + "is waiting");
     return true;
  }
  boolean drinkingState() {
     System.out.println("Customer " + this.name + " has collected coffee");
     return true;
  }
}
class Order {
  private Long id;
  public Order(Long id) {
     this.id = id;
  }
}
interface QueueOfPendingOrder {
  Queue<Order> queue = new PriorityQueue<>();
  default void add(Order order){
     queue.add(order);
  }
```

```
    default void remove(Order order) {
        queue.remove(order);
    }
}
interface QueueOfCompletedOrder {
    default void addToCompleteOrderQueue(Order order) {
        Queue<Order> queue = new PriorityQueue<>();
    }
}
```

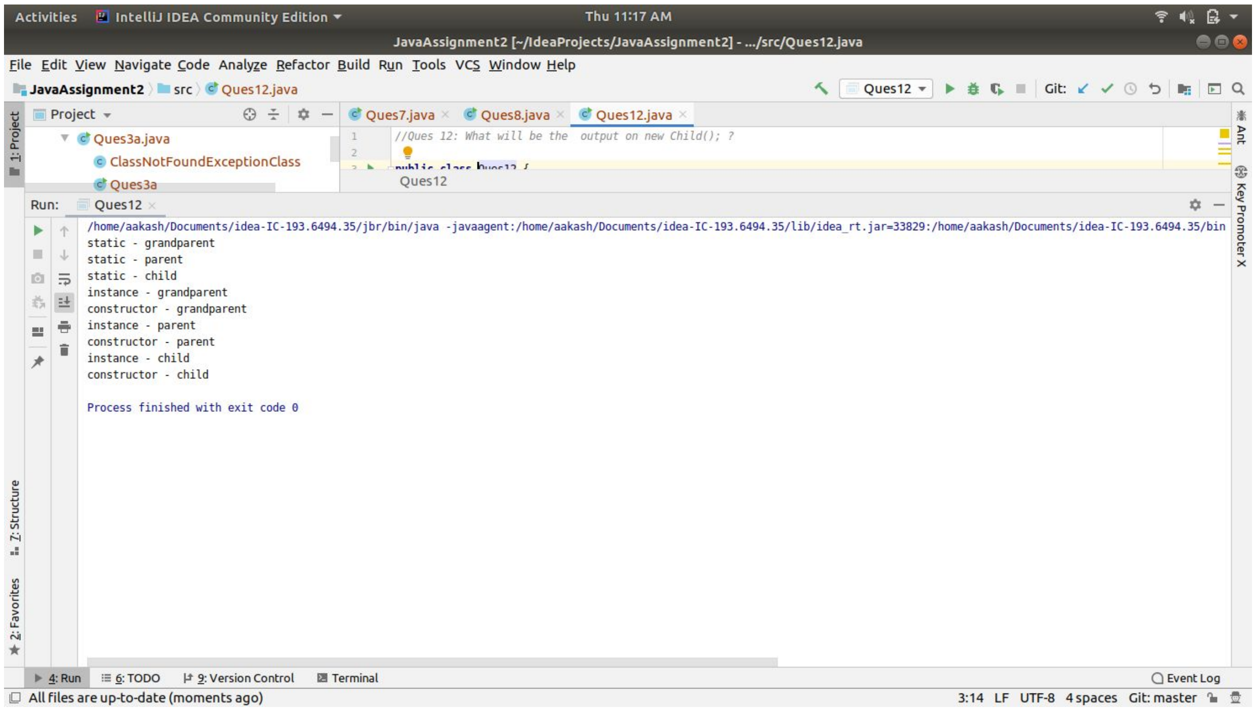
```
//Ques11: Convert the following code so that it uses nested while statements instead of for statements:
//        int s = 0;
//        int t = 1;
//        for (int i = 0; i < 10; i++)
//        {
//        s = s + i;
//        for (int j = i; j > 0; j−−)
//        {
//        t = t * (j - i);
//        }
//        s = s * t;
//        System.out.println("T is " + t);
//        }
//        System.out.println("S is " + s);

public class Ques11 {
  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);
        i++;
     }
        System.out.println("S is " + s);
  }
}
```



```java
        int j= i;
    while (j > 0)
    {
```

```
/home/aakash/Documents/idea-IC-193.6494.35/jbr/bin/java -javaagent:/home/aakash/Documents/idea-IC-193.6494.35/lib/idea_rt.jar=46063:/home/aakash/Documents/idea-IC-193.6494.35/bin
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
```

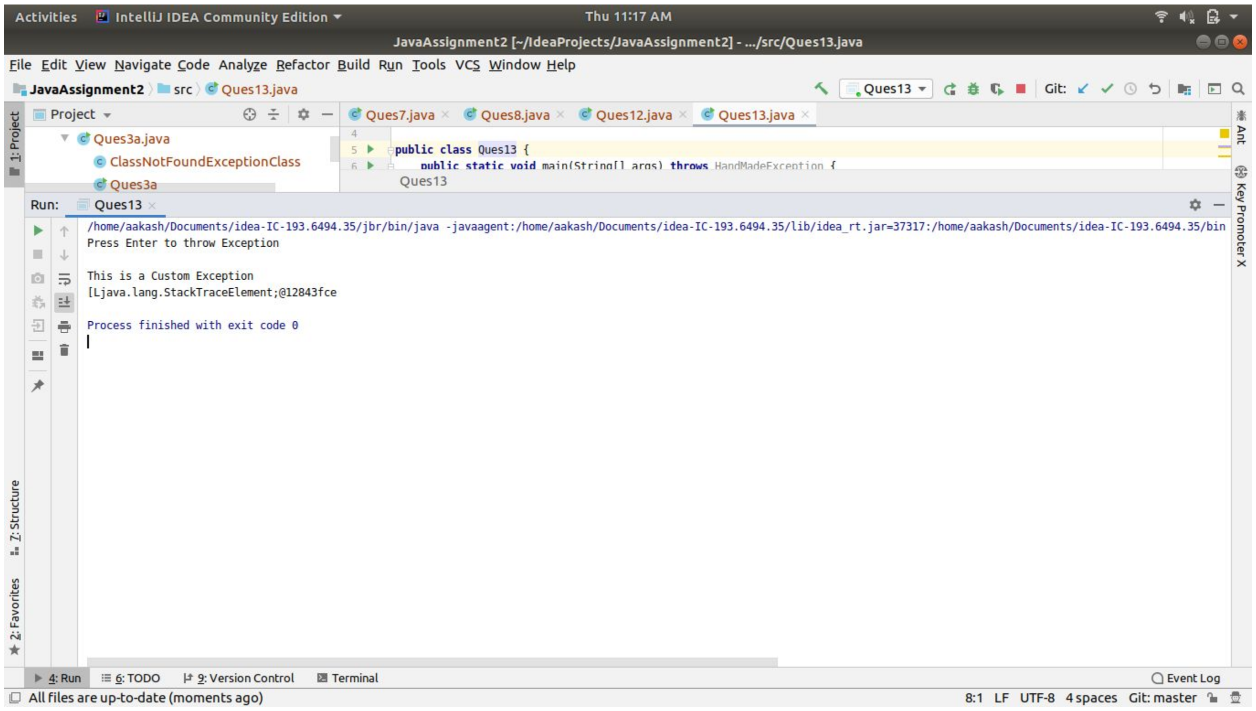
```
//Ques 12: What will be the  output on new Child(); ?

public class Ques12 {
  public static void main(String[] args) {
      Child obj = new Child();
  }

}

  class Grandparent {

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

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

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

class Parent extends Grandparent {

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

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

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

  class Child extends Parent {

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

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

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



```
//Ques 12: What will be the  output on new Child(); ?

public class Ques12 {
```

```
/home/aakash/Documents/idea-IC-193.6494.35/jbr/bin/java -javaagent:/home/aakash/Documents/idea-IC-193.6494.35/lib/idea_rt.jar=33829:/home/aakash/Documents/idea-IC-193.6494.35/bin
static - grandparent
static - parent
static - child
instance - grandparent
constructor - grandparent
instance - parent
constructor - parent
instance - child
constructor - child

Process finished with exit code 0
```

```java
//Ques 13: Create a custom exception that do not have any stack trace.

import java.util.Scanner;

public class Ques13 {
   public static void main(String[] args) throws HandMadeException {
       Scanner sc = new Scanner(System.in);

       try {
           System.out.println("Press Enter to throw Exception");
           sc.nextLine();
           throw new HandMadeException("This is a Custom Exception");
       }
       catch (HandMadeException hme){
           System.out.println(hme.getMessage());
           System.out.println(hme.getStackTrace());
       }

   }
}


class HandMadeException extends Exception{
   public HandMadeException(String message){
       super(message);
   }
}
```


Activities
IntelliJ IDEA Community Edition
Thu 11:17 AM
JavaAssignment2 [~/IdeaProjects/JavaAssignment2] - .../src/Ques13.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
JavaAssignment2 src Ques13.java
Ques13
Git:
Project
Ques3a.java
ClassNotFoundExceptionClass
Ques3a
Ques7.java
Ques8.java
Ques12.java
Ques13.java
Run:
Ques13
1: Project
7: Structure
2: Favorites
Ant
Key Promoter X
4: Run
6: TODO
9: Version Control
Terminal
Event Log
All files are up-to-date (moments ago)
8:1 LF UTF-8 4 spaces Git: master

```java
public class Ques13 {
    public static void main(String[] args) throws HandMadeException {
```

```
/home/aakash/Documents/idea-IC-193.6494.35/jbr/bin/java -javaagent:/home/aakash/Documents/idea-IC-193.6494.35/lib/idea_rt.jar=37317:/home/aakash/Documents/idea-IC-193.6494.35/bin
Press Enter to throw Exception

This is a Custom Exception
[Ljava.lang.StackTraceElement;@12843fce

Process finished with exit code 0
```