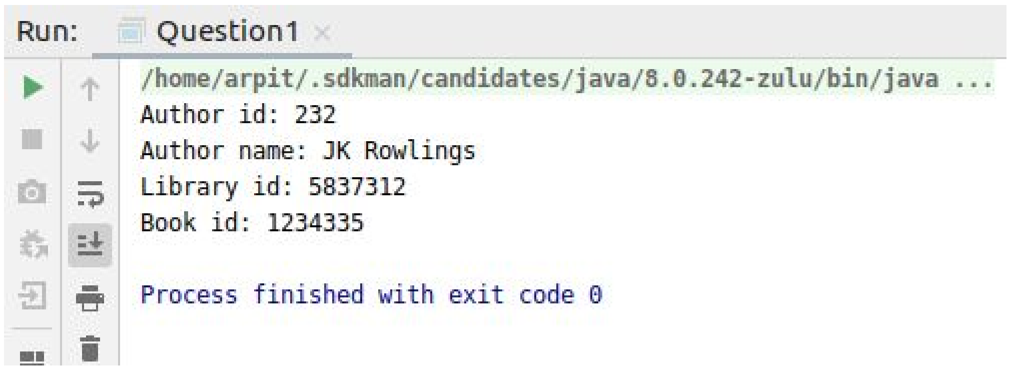
Introduction to Java 2 Exercise

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.



```
/home/arpit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Author id: 232
Author name: JK Rowlings
Library id: 5837312
Book id: 1234335

Process finished with exit code 0
```

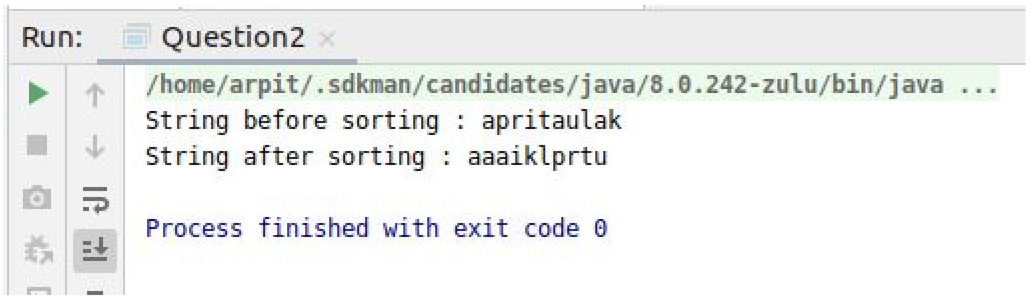
Code:-

```
package javaAssesment2;



abstract class Library {

  int libraryId = 5837312;

  abstract void libraryDetails();

}

interface Book {

  int bookId = 1234335;

  void bookDetails();

}



class Authors extends Library implements Book{
```

```java
private int authorId;

private String authorName;




public void setAuthorsDetails(int id, String name) {

    authorId = id;

    authorName = name;

}


public void setAuthorsDetails(String name, int id) {

    authorId = id;

    authorName = name;

}


void libraryDetails() {

    System.out.println("Library id: " +libraryId);

}


public void bookDetails() {

    System.out.println("Book id: " +bookId);

}


void authorDetails() {

    System.out.println("Author id: " +authorId);
```

```java
        System.out.println("Author name: " +authorName);

    }

}


public class Question1 {

    public static void main(String [] args) {

        Authors author = new Authors();

        author.setAuthorsDetails("JK Rowlings", 232);

        author.authorDetails();

        Library library = new Authors();

        library.libraryDetails();

        Book book = new Authors();

        book.bookDetails();

    }

}
```

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



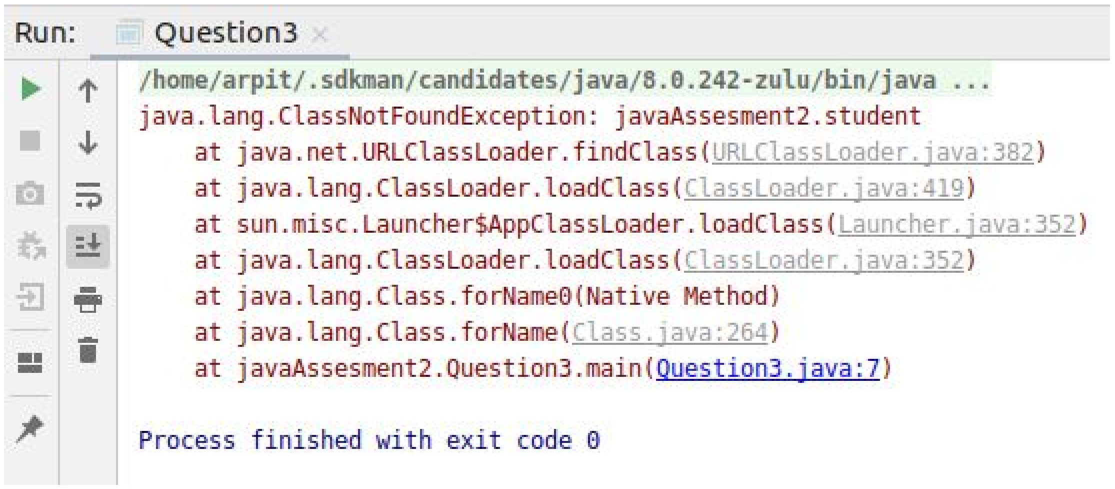
```
/home/arpit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
String before sorting : apritaulak
String after sorting : aaaiklprtu

Process finished with exit code 0
```

Code:

```java
package javaAssesment2;
```

```
public class Question2 {

  public static void main(String[] args) {

      String str = "apritaulak";

      System.out.println("String before sorting : "+str);

      String sort="";

      int j = 0;

      char temp = 0;

      char [] arr = new char[str.length()];

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

          arr[i]=str.charAt(i);

      }



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



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



              if (arr[j] > arr[i]) {

                  temp = arr[i];

                  arr[i] = arr[j];

                  arr[j] = temp;

              }

          }

      }

      System.out.println("String after sorting : "+new String(arr));
  }

}
```

3. WAP to produce NoClassDefFoundError and ClassNotFoundException exception.



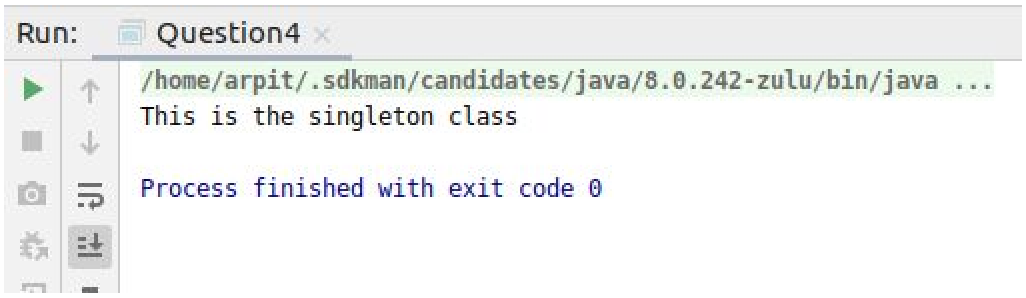
```
/home/arpit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
java.lang.ClassNotFoundException: javaAssesment2.student
    at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:419)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:352)
    at java.lang.Class.forName0(Native Method)
    at java.lang.Class.forName(Class.java:264)
    at javaAssesment2.Question3.main(Question3.java:7)

Process finished with exit code 0
```

Code:

```
package javaAssesment2;



public class Question3 {

  public static void main(String[] args) {

      try

      {

          Class.forName("javaAssesment2.student");

      }

      catch(ClassNotFoundException ex)

      {

          ex.printStackTrace();
```

```
        }
    }
}
```

4. WAP to create singleton class.



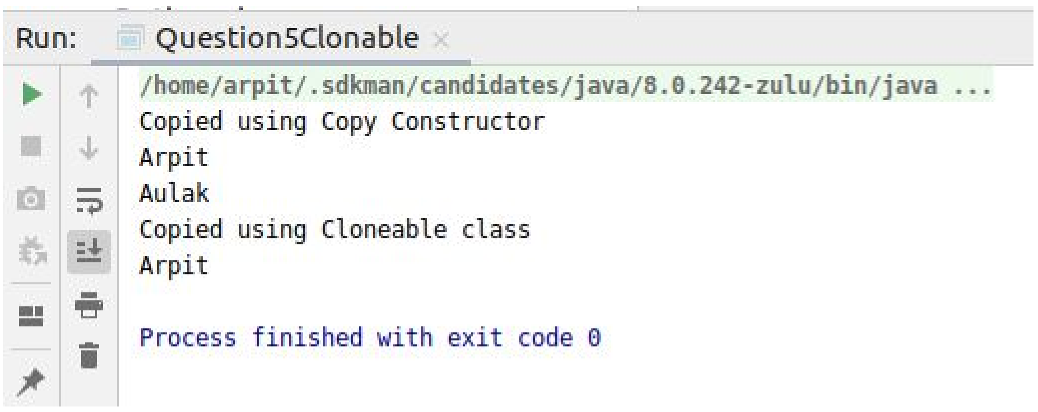
```
/home/arpit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
This is the singleton class

Process finished with exit code 0
```

Code:

```
package javaAssesment2;


public class Question4 {

    public static void main(String args[])

    {

        Singleton x = Singleton.getInstance();

        Singleton y = Singleton.getInstance();

        Singleton z = Singleton.getInstance();

    }

}


class Singleton {

    private static Singleton single_instance = null;
```

```
  public  String s;

  private Singleton()

  {

      System.out.println("This is the singleton class");

  }

  public static Singleton getInstance()

  {

      if (single_instance == null)

          single_instance = new Singleton();



      return single_instance;

  }

}
```

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



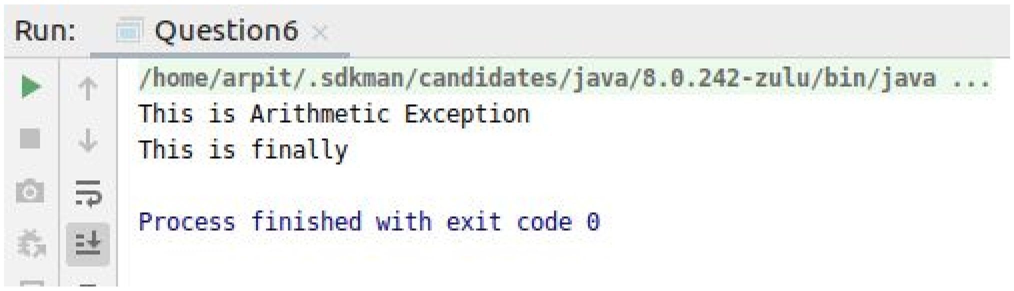
```
/home/arpit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Copied using Copy Constructor
Arpit
Aulak
Copied using Cloneable class
Arpit

Process finished with exit code 0
```

Code:

```
package javaAssesment2;
```

```
public class Question5Clonable{

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

      Copy obj = new Copy("Arpit", "Aulak");

      Copy obj2 = new Copy(obj);

      System.out.println("Copied using Copy Constructor");

      System.out.println(obj2.fName);

      System.out.println(obj2.lName);



      TryClone tc = new TryClone();

      tc.name = "Arpit";

      TryClone tc1 = (TryClone) tc.clone();

      System.out.println("Copied using Cloneable class");

      System.out.println(tc1.name.toString());



  }



}

class TryClone implements Cloneable {

  String name;



  public Object clone() throws CloneNotSupportedException{

      return super.clone();

  }
}


class Copy {

  String fName;

  String lName;


  public Copy(String f, String l)

  {

      this.fName = f;

      this.lName = l;

  }

  public Copy(Copy obj)

  {

      fName = obj.fName;

      lName = obj.lName;

  }

}
```

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



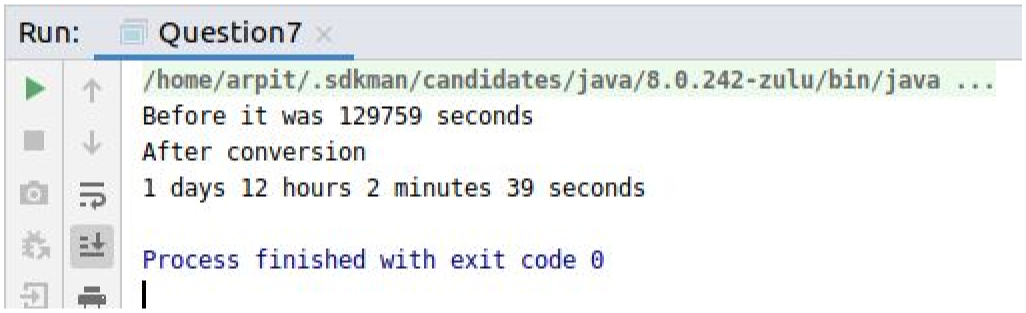
```
/home/arpit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
This is Arithmetic Exception
This is finally

Process finished with exit code 0
```

Code:

```
package javaAssesment2;


import java.util.Scanner;


public class Question6 {

  public static void main(String args[])

  {

      Scanner scn = new Scanner(System.in);

      try

      {

          int a[]=new int[5];

          a[3]=30/0;

          System.out.println(a[10]);

      }

      catch(ArrayIndexOutOfBoundsException e)

      {

          System.out.println("This is ArrayIndexOutOfBounds Exception");
```

```
    }
    catch(ArithmeticException e)
    {
      System.out.println("This is Arithmetic Exception");
    }
    finally
   {
      System.out.println("This is finally");
    }
  }
}
```

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



Run: Question7

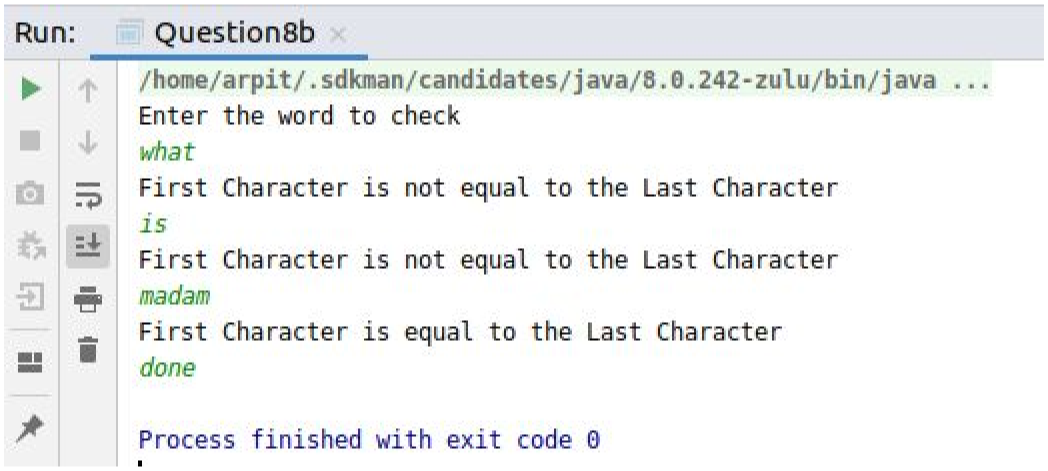
```
/home/arpit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Before it was 129759 seconds
After conversion
1 days 12 hours 2 minutes 39 seconds

Process finished with exit code 0
```

Code:

```
package javaAssesment2;


public class Question7 {
  public static void main (String[] args)
  {
        int n = 129759;

        System.out.println("Before it was "+n+" seconds");

        System.out.println("After conversion");

        Convert(n);

    }

    static void Convert(int n)

    {

        int day = n / (24 * 3600);



        n = n % (24 * 3600);

        int hour = n / 3600;



        n = n % 3600;

        int minutes = n / 60 ;

        n = n % 60;

        int seconds = n;

        System.out.println( day + " " + "days " + hour + " " + "hours " + minutes + " " + "minutes " + seconds + " " +
"seconds ");

    }

}
```

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

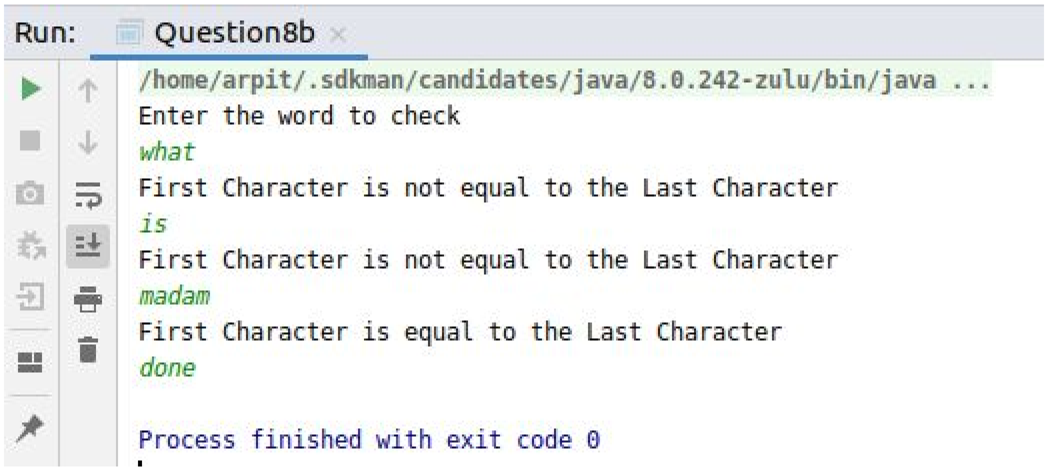


```
/home/arpit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Enter the word to check
what
First Character is not equal to the Last Character
is
First Character is not equal to the Last Character
madam
First Character is equal to the Last Character
done

Process finished with exit code 0
```

Code:

```
package javaAssesment2;



import java.util.Scanner;



public class Question8a {



  public static void main(String[] args) {

      Scanner scanner = new Scanner(System.in);

      System.out.println("Enter the word to check");

      String str = scanner.nextLine();

      while (!str.equals("done")) {

          if (str.charAt(0) == str.charAt(str.length() - 1)) {
              System.out.println("First Character is not equal to the Last Character");
          }
          else{
              System.out.println("First Character is equal to the Last Character");
          }
          str = scanner.nextLine();
      }
  }
}
```

b)do-while statement



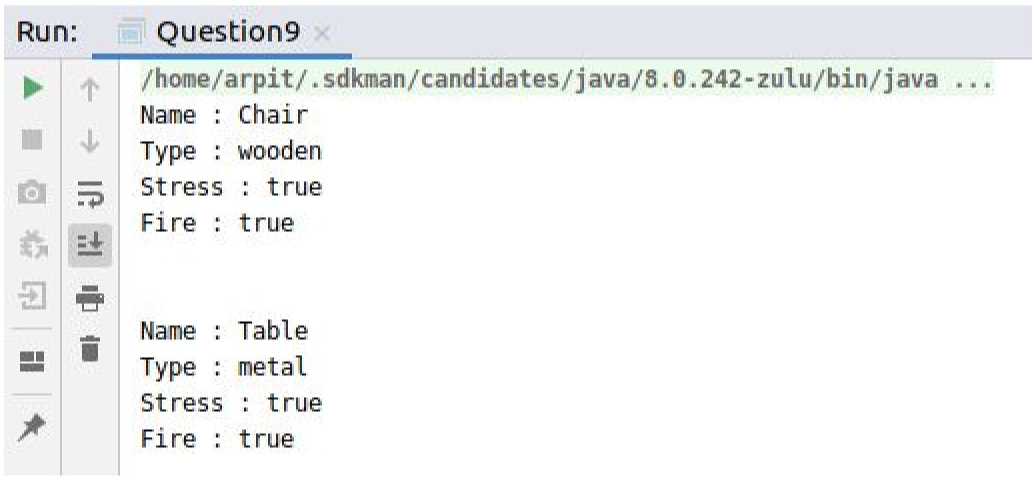
```
/home/arpit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Enter the word to check
what
First Character is not equal to the Last Character
is
First Character is not equal to the Last Character
madam
First Character is equal to the Last Character
done

Process finished with exit code 0
```

Code:

```
package javaAssesment2;


import java.util.Scanner;
```

```
public class Question8b {

  public static void main(String[] args) {

      Scanner str = new Scanner(System.in);

      System.out.println("Enter the word to check");

      String word = str.nextLine();

      do{



          if (word.charAt(0) == word.charAt(word.length()-1)){

              System.out.println("First Character is equal to the Last Character");

          }

          else{



              System.out.println("First Character is not equal to the Last Character");

          }



          word =str.nextLine();

      }while(!word.equals("done"));



  }

}
```

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.



```
/home/arpit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Name : Chair
Type : wooden
Stress : true
Fire : true


Name : Table
Type : metal
Stress : true
Fire : true
```

Code:

```java
package javaAssesment2;



public class Question9 {

  public static void main(String[] args) {

      chair ch = new chair("wooden", true, true);

      table tb = new table("metal", true, true);

      ch.getDetails();

      tb.getDetails();

  }

}



abstract class furniture {

  public abstract boolean checkStress();

  public abstract boolean checkfire();
```

```
    public abstract void getDetails();

}


class chair extends furniture{

    String type;

    boolean fire;

    boolean stress;



    public chair(String t, boolean f, boolean s)

    {

        this.type=t;

        this.fire=f;

        this.stress=s;

    }



    public String getType() {

        return type;

    }

    public boolean checkStress()

    {

        return stress;

    }

    public boolean checkfire() {

        return fire;
    }

    public void getDetails() {

        System.out.println("Name : Chair");

        System.out.println("Type : "+getType());

        System.out.println("Stress : "+checkStress());

        System.out.println("Fire : "+checkfire());

        System.out.println("\n");

    }

}


class table extends furniture{

    String type;

    boolean fire;

    boolean stress;


    public table(String t, boolean f, boolean s)

    {

        this.type=t;

        this.fire=f;

        this.stress=s;

    }


    public String getType() {

        return type;
    }
    public boolean checkStress()
    {
        return stress;
    }
    public boolean checkfire() {
        return fire;
    }
    public void getDetails() {
        System.out.println("Name : Table");
        System.out.println("Type : "+getType());
        System.out.println("Stress : "+checkStress());
        System.out.println("Fire : "+checkfire());
        System.out.println("\n");
    }
}
```

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

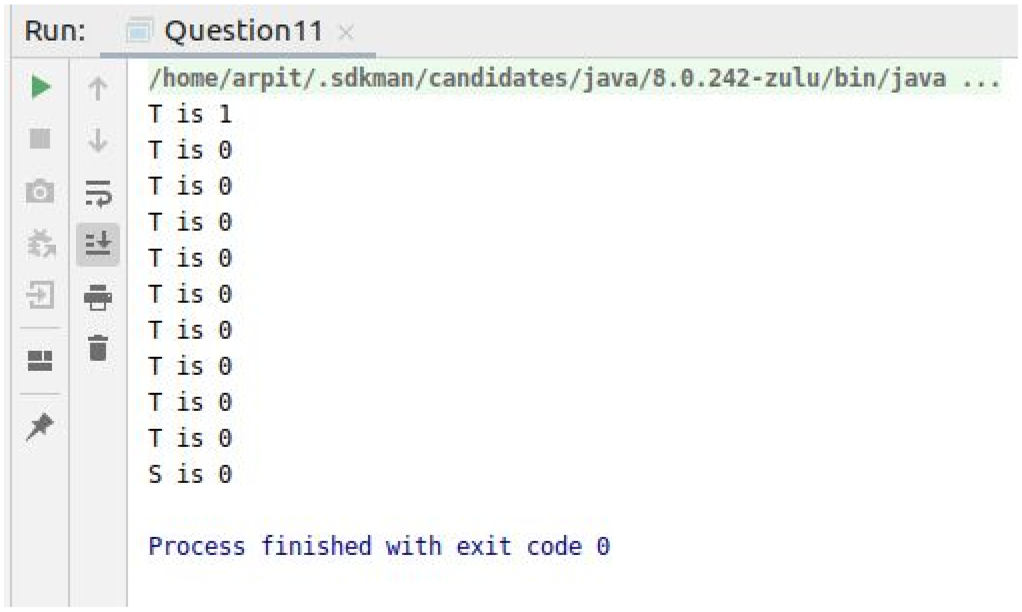
 - Places a notification that order for token is ready

Code Design:

```
class Customer{

  String name;

  long contact;

  public void wait_for_coffee(){}

  public void collects_coffee(){}



}



class Cashier{
  String name;

  String Empid;

  public int take_order(double cash)

  {

      return 0;

  }

  public void placeOrderToQueue(Order o){}

}

class Order{

  int order_id;

  String coffee_type;

}

class Barista{

  public void Prepare_coffee(Order o) {}

  public void addCompleteOrder(Order o) {}

  public void notifyCompleteOrder(Order o) {}

}

class PendingOrderQueue

{

  public void addOrder(Order o){}

  public void getNextOrder(Order o){}

}

class CompleteOrders{

  public void getcompleteOrders(){}
```

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);
```



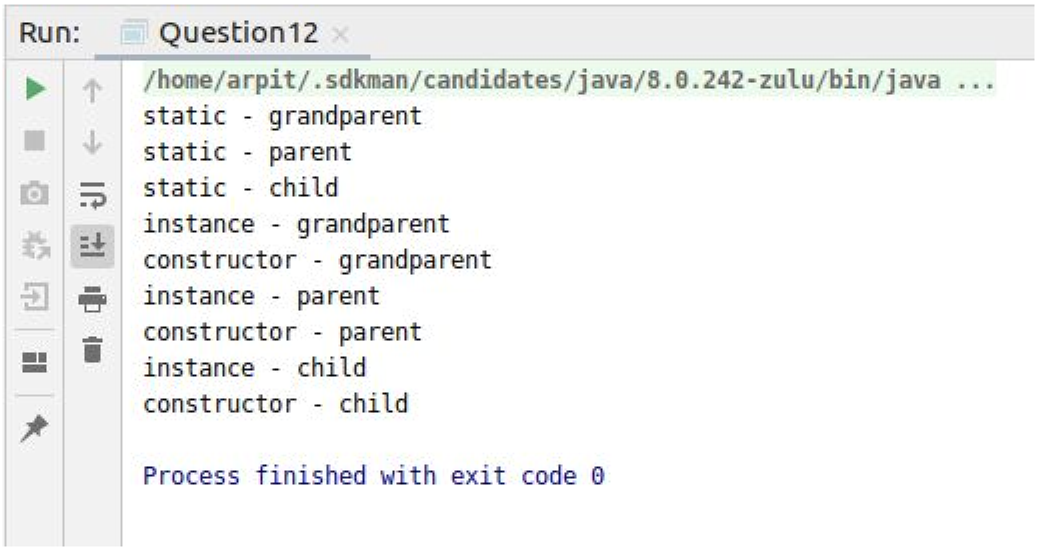
```
/home/arpit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
T is 1
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
S is 0

Process finished with exit code 0
```

Code:

```
package javaAssesment2;


public class Question11 {

  public static void main(String[] args) {

      int s = 0;

      int t = 1;

      int i = 0;

      int j;

      while (i<10)

      {

          s = s + i;

          j=i;

          while (j>0)

          {
              t = t * (j - i);

              j--;

          }

          s = s * t;

          i++;

          System.out.println("T is " + t);

      }

      System.out.println("S is " + s);

  }

}
```

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");
        }
    }
```



```
/home/arpit/.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
```

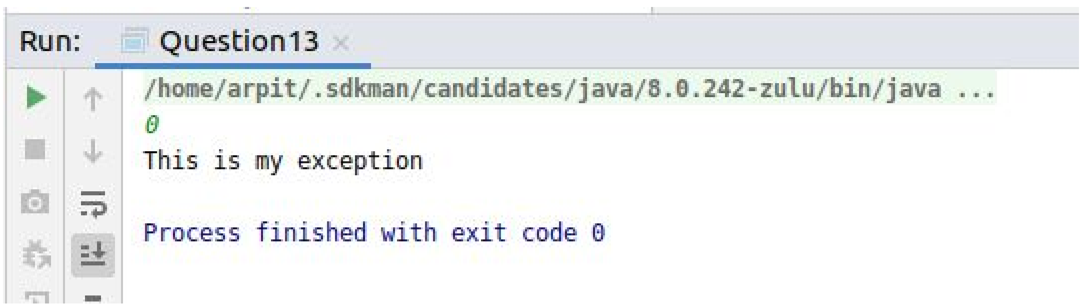
Ans. In this if i make a child class object and call it then the following will happen:

1. All the static blocks in the classes will execute first.
2. After that their other blocks and constructors will execute.

Code:

```
public class Question12 {
  public static void main(String[] args) {
     Child c= new Child();
  }
}
```

Q13. Create a custom exception that do not have any stack trace.



```
/home/arpit/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
0
This is my exception

Process finished with exit code 0
```

Code:

```
package javaAssesment2;

import java.util.Scanner;

class MyException extends Exception{
  public MyException(String msg)
  {
      super(msg);
  }
}


public class Question13 {
  public static void main(String[] args)throws MyException {
      Scanner sc = new Scanner(System.in);
      int a = sc.nextInt();
      try
      {
          if(a==0)
          {
              throw new MyException("This is my exception");
          }
      }
      catch(MyException e)
      {
          System.out.println(e.getMessage());
      }
  }
}
```