Q1. 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-

```
interface Account {
  public void account();
}

class Librarian implements Account {
  String name;
  int id;

  public void account() {
     String id;
     String pass;
  }

  public String getName() {
     return name;
  }

  public void setName(String name) {
     this.name = name;
  }

  public int getId() {
     return id;
  }

  public void setId(int id) {
     this.id = id;
  }
}

class User implements Account {
  int id;
  String name;

  public int getId() {
     return id;
  }

  public void setId(int id) {
     this.id = id;
  }

  public String getName() {
     return name;
  }
```

```
    public void setName(String name) {
        this.name = name;
    }

    public void account() {
        String id;
        String pass;
    }
}

class Book {
    String author;
    String title;
    double price;

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }
}

public class Q1 {
    public static void bookAssigned() {
        Librarian libr = new Librarian();
        Book book = new Book();
        User user = new User();
        libr.setName("Anmol Dua");
        libr.setId(1001);
        book.setAuthor("Rajesh");
        book.setTitle("Phool Aur Kante");
        book.setPrice(350.5);
```

```
        user.setName("Tushar Garg");
        user.setId(4000);
        System.out.println("Book Assigned to - ");
        System.out.println("Name : " + user.getName() + "; ID : " + user.getId());
        System.out.println("Assigned by :- " + libr.getName() + "; ID : " + libr.getId());
        System.out.println("Book details :-");
        System.out.println(" Book : " + book.getTitle() + "; Author : " + book.getAuthor() + "; Price : " +
book.getPrice());
    }

    public static void main(String[] args) {
        Q1 obj = new Q1();
        obj.bookAssigned();
    }
}
```

Output-

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Book Assigned to -
Name : Tushar Garg; ID : 4000
Assigned by :- Anmol Dua; ID : 1001
Book details :-
 Book : Phool Aur Kante; Author : Rajesh; Price : 350.5

Process finished with exit code 0
```

Q2. WAP to sorting string without using string Methods?.

Code

```
public class q2
{
    public static void main(String[] args)
    {

        String original = " i live in faridabad";
        int j = 0;
        char temp = 0;
        char[] tush = original.toCharArray();
        for (int i = 0; i < tush.length; i++)
        {
            for (j = 0; j < tush.length; j++)
            {
                if (tush[j] > tush[i])
                {
                    temp = tush[i];
                    tush[i] = tush[j];
                    tush[j] = temp;
                }
            }
        }
        for (int k = 0; k < tush.length; k++)
        {
            System.out.println(tush[k]);
        }
    }
}
```

Output

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...



a
a
a
b
d
d
e
f
i
i
i
i
l
n
r
v

Process finished with exit code 0
```

Q3 WAP to produce NoClassDefFoundError and ClassNotFoundException exception.

Code1- ClassNotFoundException

```
import javax.naming.Name;

public class Q3 {
    public static void main(String args[]) {
        try
        {
            Class.forName("Tushar Garg");
        }
        catch (ClassNotFoundException ex)
        {
            System.out.println(ex);
            System.out.println("This is catch box");
        }
    }
}
```

Output

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
java.lang.ClassNotFoundException: Tushar Garg
This is catch box

Process finished with exit code 0
```

Code2-NoClassDefFoundError When Class is present

```
class q3 {
    void tushar() {
        System.out.println("This is Tushar");
    }
}

class mohit{
    public static void main(String args[])
    {
        q3 obj = new q3();
        obj.tushar();
    }
}
```

Output

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
This is Tushar

Process finished with exit code 0
```

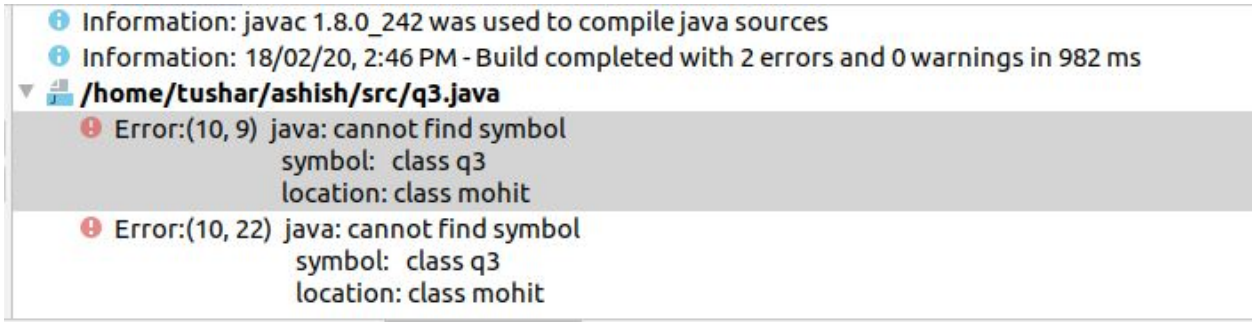
2.- When class is not present

Code

```
/*class q3 {
    void tushar() {
        System.out.println("This is Tushar");
    }
}*/

class mohit{
    public static void main(String args[])
    {
        q3 obj = new q3();
        obj.tushar();
    }
}
```

Output



Q4 WAP to create singleton class.

Code-

```
public class Practice {
    private static Practice myObj;
    static {
        myObj = new Practice();
    }
    private Practice() {
    }
    public static Practice getInstance() {
        return myObj;
    }
    public void testMe() {
        System.out.println("Hey.... it is working!!!");
    }
    public static void main(String a[]) {
        Practice ms = getInstance();
        ms.testMe();
    }
}
```

Ouput

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Hey.... it is working!!!

Process finished with exit code 0
```

Q5 WAP to show object cloning in java using cloneable and copy constructor both.

Code-

```
import java.io.*;
class Test
{
        int x, y;Test()
        {
            x = 10;
            y = 20;
        }
}

class Main
{
        public static void main(String[] args)
        {
            Test ob1 = new Test();
            System.out.println(ob1.x + " " + ob1.y);
            Test ob2 = ob1;
            ob2.x = 100;

            System.out.println(ob1.x + " " + ob1.y);
            System.out.println(ob2.x + " " + ob2.y);
        }
}
```

Output

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
10 20
100 20
100 20

Process finished with exit code 0
```

Q6 WAP showing try, multi-catch and finally blocks.

Code-

```
import java.io.*;
import java.nio.channels.ScatteringByteChannel;
import java.util.Scanner;

public class MultipleCatch {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        try {
            int num = Integer.parseInt(input.nextLine());
            if (500 % num == 0) {
                System.out.println("This is a factor of 500");
            }
        }
        catch (ArithmeticException e){
            System.out.println("Catch 1");
            System.out.println(e);
        }
        catch (NumberFormatException e){
            System.out.println("Catch 2");
            System.out.println(e);
        }
        finally {
            System.out.println("Finally");
        }
    }
}
```

Output-

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
0
Catch 1
java.lang.ArithmeticException: / by zero
Finally

Process finished with exit code 0
```

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
5.25
Catch 2
java.lang.NumberFormatException: For input string: "5.25"
Finally

Process finished with exit code 0
```

Q7 WAP to convert seconds into days, hours, minutes and seconds.

Code

```
import java.util.Scanner;
class convertseconds {
    public static void main(String[] args) {
        Scanner tush = new Scanner(System.in);
        System.out.println("Enter Seconds:");
        int sec = tush.nextInt();
        int day = sec / (24 * 3600);
        sec = sec % (24 * 3600);
        int hour = sec / 3600;
        sec %= 3600;
        int minutes = sec / 60;
        sec %= 60;
        int seconds = sec;
        System.out.println(day + " " + "days \n" + hour + " " + "hours \n" + minutes + " "
                + "minutes \n" + seconds + " " + "seconds ");
    }
}
```

Output

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Enter Seconds:
8974564
103 days
20 hours
56 minutes
4 seconds

Process finished with exit code 0
```

Q8 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

A - Code

```
import java.util.*;
    public class WordDone
    {
        public static void main(String[] args)
        {
            Scanner input = new Scanner(System.in);
            String word = input.next();
            while(!word.equals("done"))
            {
                if(word.charAt(0) == word.charAt(word.length()-1))
                {
                    System.out.println("First and last words are equal.");
                }
                else
                    {
                    System.out.println("First and last words are not equal.");
                }
                word = input.next();
            }
        }
    }
```

Output

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
jnevjn
First and last words are not equal.
noun
First and last words are equal.
done

Process finished with exit code 0
```

B - Code

```java
import java.util.*;
    public class Worddone
    {
        public static void main(String[] args) {
            Scanner input = new Scanner(System.in);
            String word = input.next();
            do {
                if (word.charAt(0) == word.charAt(word.length() - 1))
                {
                    System.out.println("First and last words are equal.");
                } else
                    {
                    System.out.println("First and last words are not equal.");
                }
                word = input.next();
            }
            while(!word.equals("done"));
        }
    }
```

Output -

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
fwye
First and last words are not equal.
prop
First and last words are equal.
done

Process finished with exit code 0
```

Q9 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.

Code-

```
import java.util.Scanner;

interface Furniture {
  public void stressTest();

  public void fireTest();
}

abstract class Table implements Furniture {
  public abstract String tableType();
}

abstract class Chair implements Furniture {
  public abstract String chairtype();
}

class MetalTable extends Table {
  public void stressTest() {
    System.out.println("Stress Test Passed....");
  }

  public void fireTest() {
    System.out.println("Fire Test Passed....");
  }

  public String tableType() {
    String s = "Metal Table";
    return s;
  }
}

class WoodenTable extends Table {
  public void stressTest() {
    System.out.println("Stress Test Passed....");
  }

  public void fireTest() {
    System.out.println("Fire Test failed.... ");
  }

  public String tableType() {
    String s = "Wooden Table";
    return s;
  }
}
```

```
class MetalChair extends Chair {
  public void stressTest() {
     System.out.println("Stress Test Passed");
  }

  public void fireTest() {
     System.out.println("Fire Test Passed");
  }

  public String chairtype() {
     String s = "Metal Chair";
     return s;
  }
}

class WoodenChair extends Chair {
  public void stressTest() {
     System.out.println("Stress Test Failed");
  }

  public void fireTest() {
     System.out.println("Fire test Failed");
  }

  public String chairtype() {
     String s = "Wooden Chair";
     return s;
  }
}

public class Q11{
  public static void main(String[] args) {
     Chair chair = null;
     Table table = null;
     Scanner input = new Scanner(System.in);

     System.out.println("Enter Furniture type:- table/chair");
     String str1 = input.nextLine();

     if (str1.equals("chair")) {
        System.out.println("Enter Chair type:- wooden/metal");
        String str = input.nextLine();
        if (str.equals("wooden")) {
           chair = new WoodenChair();
           System.out.println(chair.chairtype());
           chair.fireTest();
           chair.stressTest();
        } else if (str.equals("metal")) {
           chair = new MetalChair();
           System.out.println(chair.chairtype());
```

```
            chair.fireTest();
            chair.stressTest();
        } else {
            System.out.println("Wrong input...");
        }

    } else if (str1.equals("table")) {
        System.out.println("Enter Table type:- wooden/metal");
        String str = input.nextLine();
        if (str.equals("wooden")) {
            table = new WoodenTable();
            System.out.println(table.tableType());
            table.fireTest();
            table.stressTest();
        } else if (str.equals("metal")) {
            table = new MetalTable();
            System.out.println(table.tableType());
            table.fireTest();
            table.stressTest();
        } else {
            System.out.println("Wrong input...");
        }

    } else {
        System.out.println("Wrong Input...");
    }

  }
}
```

Output-

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Enter Furniture type:- table/chair
table
Enter Table type:- wooden/metal
wooden
Wooden Table
Fire Test failed....
Stress Test Passed....

Process finished with exit code 0
```

Q11 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);
```

Ans-

```
int s=0;
int t=1;
int i=0;
while(i< 10)
{
s=s+i;
int j;
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);
```

Q12 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");
    }
}
```

Output

1. static - grandparent
2. static-parent
3. Static-child
4. instance-grandparent
5. constructor-grandparent
6. instance-parent
7. constructor-parent
8. instance-child
9. constructor-child

-

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

Code-

```
public class Q13{
    public static void main(String[] args) {
        int accBal = 500;
        int transfer = 600;
        try{
            if (transfer > accBal) {
                throw new MyException();
            }
        }
        catch (MyException e)
        {
            e.showError();
        }
    }
}

    class MyException extends Exception{
        void showError(){
            System.out.println("Error : Amount entered is greater then account balance.");
        }
    }
```

Output

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Error : Amount entered is greater then account balance.

Process finished with exit code 0
```