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.public void author_name();

```
package assignment2;
import java.util.Scanner;
interface books_issue
{
   public void issued_on();
   public void return_date();
}
interface book_details
{
public void book_name();
public void author_name();
}
interface person_details
{
    public void person_name();
}
class person implements book_details ,books_issue,person_details
{
    String issue_name,issue_date,author_nm,date_ret,b_name,person_nm;
    Scanner scan=new Scanner(System.in);
    public void issued_on(){
        System.out.println("Enter book issue date");
        issue_date=scan.nextLine();
    }
    public void return_date()
    {
        System.out.println("Enter book return date");
        date_ret=scan.nextLine();
    }
    public void book_name()
    {
        System.out.println("Enter book name");
        b_name=scan.nextLine();
    }
    public void author_name()
    {
        System.out.println("Enter author name");
        author_nm=scan.nextLine();
    }
    public void person_name()
    {
        System.out.println("Enter author name");
       person_nm=scan.nextLine();
    }
    public void show()
    {
        System.out.println("Person name is: "+person_nm);
        System.out.println("Book name is: "+b_name);
        System.out.println("Author name is: "+author_nm);
        System.out.println("Date of issue is: "+issue_date);
        System.out.println("Date of return is: "+date_ret);
    }
}
class main1
{
    public static void main(String[] args) {
        int a=0;
        Scanner scan = new Scanner(System.in);
        person p = new person();
        while (a != 7) {
            System.out.println("Enter 1 for insertion of name");
```

```
            System.out.println("Enter 2 for insertion of book name");
            System.out.println("Enter 3 for insertion of author name");
            System.out.println("Enter 4 for insertion of issue_date");
            System.out.println("Enter 5 for date of return");
            System.out.println("Enter 6 to show record");
            System.out.println("Enter 7 for exit");
             a = scan.nextInt();
            if (a == 1) {
                p.person_name();
            } else if (a == 2) {
                p.book_name();
          } else if (a == 3) {
                p.author_name();
            } else if (a == 4) {
                p.issued_on();
                ;
            } else if (a == 5) {
                p.return_date();
            } else if (a == 6) {
                p.show();
            }
        }
    }
}
```

```
Enter 5 for date of return
Enter 6 to show record
Enter 7 for exit
6
Person name is: Siddharth
Book name is: Kh
Author name is: dfasd
Date of issue is: 12-3-2020
Date of return is: 15-4-2020
```

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

```
package assignment2;
public class as2q2 {
    public static void main(String[] args) {
        String abc="siddharthbhatia";
        int arr[]=new int[20];
        for(int i=0;i<abc.length();i++)
        {
            char a=abc.charAt(i);
            int assci=(int) a;
            arr[i]=assci;
        }
        int temp;
        for(int i=0;i<abc.length();i++)
        {
            for (int j = i + 1; j < abc.length(); j++) {
                if (arr[i] > arr[j]) {
                    temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
        int arr2[]=new int[50];
        for(int i=0;i<abc.length();i++)
```

```
        {
            int a=arr[i];
            char asscii=(char)a;
            arr2[i]=asscii;
        }
        for(int i=0;i<abc.length();i++)
        {
            String str=Character.toString(arr2[i]);
            System.out.print(" "+str);
        }
    }
}
```

```
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intellij-idea-community/208/lib/idea_rt.jar=33815:/snap/intellij-idea-community/2
  .encoding=UTF-8 -classpath /home/siddharth/IdeaProjects/jvm_assignment/out/production/jvm_assignment assignment2.as2q2
 a a a b d d h h h i i r s t t
Process finished with exit code 0
```

6: TODO Terminal 0: Messages

q-3 WAP to produce NoClassDefFoundError and ClassNotFoundException exception.
No clas def found error

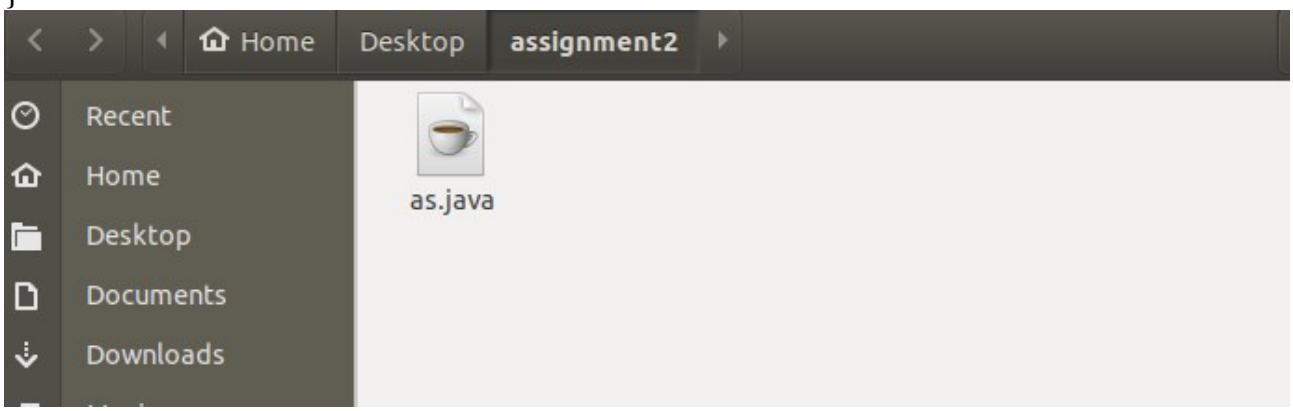
```
package assignment2;
public class as2q3 {
    public static void main(String[] args) {
        try
        {
            Class.forName("Siddharth");
        }
        catch (ClassNotFoundException ex)
        {
            ex.printStackTrace();
        }
    }
}
```

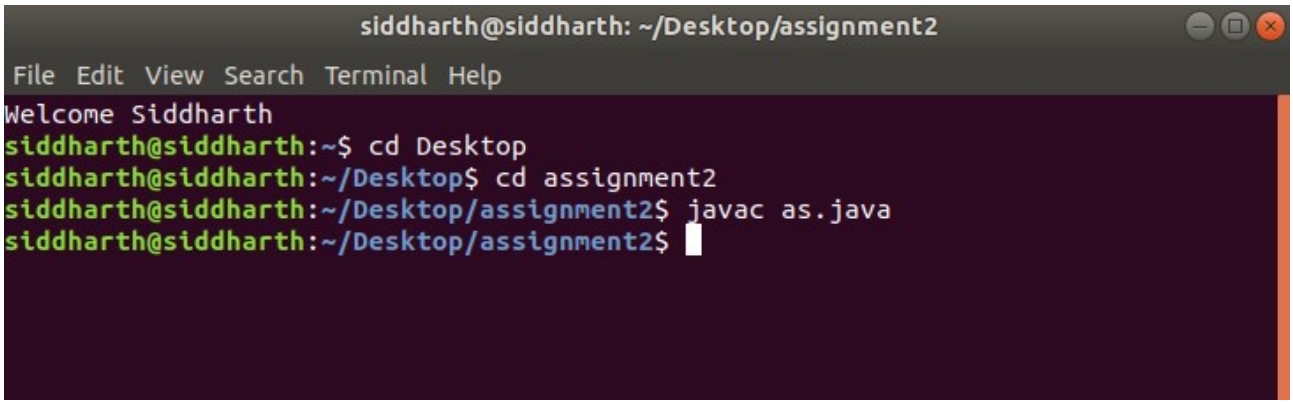
```
  .encoding=UTF-8 -classpath /home/siddharth/IdeaProjects/jvm_assignment/out/production/jvm_assignment assignment2.as2q3
java.lang.ClassNotFoundException: Siddharth <2 internal calls>
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:521)
    at java.base/java.lang.Class.forName0(Native Method)
    at java.base/java.lang.Class.forName(Class.java:315)
    at assignment2.as2q3.main(as2q3.java:7)

Process finished with exit code 0
```

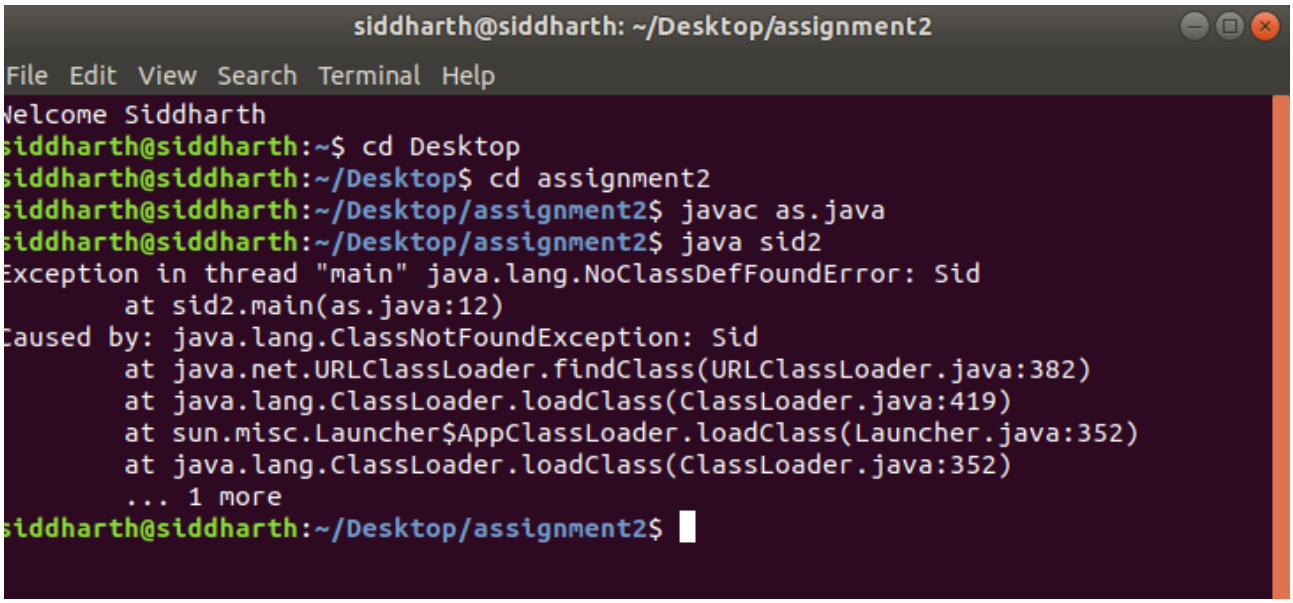
No Class Def found

```
class Sid
{
   void greeting()
   {
      System.out.println("hello world");
   }
}

class sid2 {
   public static void main(String args[])
   {
      Sid geeks = new Sid();
```

```
        geeks.greeting();
    }
}
```







q-4 WAP to create singleton class.

```
package assignment2;
class singla
{
    public static  void main(String [] args)
    {
        as2q4 obj=as2q4.getInstance();
    }
}
class as2q4
{
    static as2q4 obj=new as2q4();
    private as2q4()
    {
    }
    public static as2q4 getInstance()
    {
        return obj;
    }
}
```

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

```
package collecton_prac;
class Cat2{
    int j;
    Cat2(int j)
    {
        this.j=j;
    }
}
class Dog2 implements  Cloneable{
    Cat2 c;
    int i;
    Dog2(Cat2 c,int i)
    {
        this.c=c;
        this.i=i;
    }
    Dog2(Dog2 x)
    {
        this.c=x.c;
        this.i=x.i;
    }
    public Object clone() throws CloneNotSupportedException
    {
        return super.clone();
    }
}
public class shallow_cloning {
    public static void main(String[] args) throws CloneNotSupportedException {
        Cat2 c=new Cat2(20);
        Dog2 d1=new Dog2(c,10);
        System.out.println(d1.i+"...."+d1.c.j);
        Dog2 d2=(Dog2)d1.clone();
        d2.i=888;
        d2.c.j=999;
        System.out.println(d1.i+"...."+d1.c.j);
        Dog2 d3=new Dog2(d2);
        System.out.println("Copy constructor: "+d3.i + "...."+d3.c.j);
    }
}
```

```
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intellij-idea-community/208/lib/idea_rt.jar=44801:/snap/intellij-idea-community/208/bin -Dfile
  .encoding=UTF-8 -classpath /home/siddharth/IdeaProjects/jvm_assignment/out/production/jvm_assignment collecton_prac.shallow_cloning
10....20
10....999
Copy constructor: 888....999

Process finished with exit code 0
```

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

```
package assignment2;
public class as2q6 {
    public static void main(String[] args) {
        int arr[]=new int[5];
        try{
            System.out.println(10/0);
            for(int i=0;i<6;i++)
            {
                arr[i]=i;
            }
        }
        catch(ArithmeticException e)
        {
          System.out.println("Can divide by zero");
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Insertion more than required");
        }
        finally
        {
            System.out.println("Cleaning of try block done by me");
        }
    }
}
```

```
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intellij-idea-
  .encoding=UTF-8 -classpath /home/siddharth/IdeaProjects/jvm_assignment/out/p
Can divide by zero
Cleaning of try block done by me

Process finished with exit code 0
```

q-7 WAP to convert seconds into days, hours, minutes and seconds

```
package assignment2;
public class as2q7 {
    static void Convert ( int n)
    {
        int day = n / (24 * 3600);
        n = n % (24 * 3600);
        int hour = n / 3600;
        n %= 3600;
        int minutes = n / 60;
        n %= 60;
        int seconds = n;
        System.out.println(day + " " + "days " + hour + " " + "hours " + minutes + " " +
"minutes " + seconds + " " + "seconds ");
    }
    public static void main(String[] args) {
        int n = 129600;
        Convert(n);
    }
```

```
   /snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intellij-idea-community
     .encoding=UTF-8 -classpath /home/siddharth/IdeaProjects/jvm_assignment/out/production
   1 days 12 hours 0 minutes 0 seconds

   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

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.

```
package assignment2;
abstract class furniture {
    public abstract boolean checkStress();
    public abstract boolean checkfire();
}
class table extends furniture{
    String type;
    boolean fire;
    boolean stress;
    public String getType() {
        return type;
    }
    public table(String t, boolean f, boolean s)
    {
        this.type=t;
        this.fire=f;
        this.stress=s;
    }
    public boolean isFire() {
        return fire;
    }
    public boolean isStress() {
        return stress;
    }
    public boolean checkStress()
    {
        return stress;
    }
    public boolean checkfire() {
        return fire;
    }
}
class chair extends furniture{
    String type;
    boolean fire;
    boolean stress;
    public String getType() {
        return type;
    }
    public chair(String t, boolean f, boolean s)
    {
        this.type=t;
        this.fire=f;
        this.stress=s;
    }
```

```java
    public boolean isFire() {
        return fire;
    }
    public boolean isStress() {
        return stress;
    }
    public boolean checkStress()
    {
        return stress;
    }
    public boolean checkfire() {
        return fire;
    }
}
public class as2q9 {
    public static void main(String[] args) {
        chair ch = new chair("wooden", true, true);
        table tb = new table("metal", true, true);
        System.out.println("For CHAIR--");
        System.out.println("Type is: "+ch.type);
        System.out.println("Stress test : "+ch.checkStress());
        System.out.println("Fire test : "+ch.checkfire());
        System.out.println("*******************************");
        System.out.println("For TABLE--");
        System.out.println("Type is: "+tb.type);
        System.out.println("Stress test: "+tb.checkfire());
        System.out.println("Fire test: "+tb.checkStress());
    }
}
```

```
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intellij-idea-community/208/lib/idea_rt.jar=33171:/snap/intellij-idea-community/208/bin -Dfile
  .encoding=UTF-8 -classpath /home/siddharth/IdeaProjects/jvm_assignment/out/production/jvm_assignment assignment2.as2q9
For CHAIR--
Type is: wooden
Stress test : true
Fire test : true
*******************************
For TABLE--
Type is: metal
Stress test: true
Fire test: true

Process finished with exit code 0
```

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

```
package assignment2;
class customer extends cashier {
    public String name;
    public  void cust() { }
    void payment() { }
    void ready() { }
}
class cashier {
    public String order;
    void order_add(){}
}
class Barista {
    String coffe_type;
    void prepration() { }
}
```

q-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 assignment2;
public class as2q11 {
    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;
            System.out.println("T is " + t);
            i++;
        }
        System.out.println("S is " + s);
    }
}
```

```
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intellij-idea-community/208/lib/idea_rt.jar=38099:/snap/intellij-idea-community/
  .encoding=UTF-8 -classpath /home/siddharth/IdeaProjects/jvm_assignment/out/production/jvm_assignment assignment2.as2q11
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
```

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");
    }
  }
```

```
package assignment2;
 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");
    }
}
public class as2q12 {
    public static void main(String[] args) {
     Child c=new Child();
    }
}
```

```
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/
  .encoding=UTF-8 -classpath /home/siddharth/IdeaProjects/jv
static - grandparent
static - parent
static - child
instance - grandparent
constructor - grandparent
instance - parent
constructor - parent
instance - child
constructor - child
```

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

```java
package assignment2;
class as2q13 extends RuntimeException {
    as2q13(String s)
    {
        super(s);
    }
}
class cust_excpt {
    public static void main(String[] args) {
        int age = 17;
        if (age < 18) {
            throw new as2q13("\nYou are less than 18");
           // Throwable("String message, Throwable cause, boolean
enableSuppression,boolean writableStackTrace)
        }
    }
}
```

```
/snap/intellij-idea-community/208/jbr/bin/java -javaagent:/snap/intellij-idea-community/208/lib/idea_rt.jar=37277:/snap/intellij-idea-commun
  .encoding=UTF-8 -classpath /home/siddharth/IdeaProjects/jvm_assignment/out/production/jvm_assignment assignment2.cust_excpt
Exception in thread "main" assignment2.as2q13:
You are less than 18
    at assignment2.cust_excpt.main(as2q13.java:14)

Process finished with exit code 1
```