



Object Oriented Programming Language

LAB FILE

SUBMITTED BY:
TO:

Vamika Mahajan

B TECH. CSE CC&VT B6
(2ND YEAR, 3rd Semester)

500096495

R2142211138

SUBMITTED

S. CHRISTALIN NELSON

ASSISTANT PROFESSOR

SYSTEMICS CLUSTER

SOCS, UPES

INDEX

EXPERIMENT NUMBER	EXPERIMENT NAME	Page No.
1	Experiment 1- History of Java	3
2	Experiment 2-Basics of Java	7
3	Experiment 3- Arrays	13
4	Experiment 4- Inheritance	19
5	Experiment 5- Interfaces	25
6	Experiment 6-Packages	31
7	Experiment 7-Exceptions	44
8	Experiment 8-Strings	53
9	Experiment 9-Threads	59
10	Experiment 10- Collections	63
11	Experiment 11- JDBC	67
12	Experiment 12- Servlet and JSP	78

Experiment 1- History of Java

Q.1 VERSIONS OF JAVA

Java is an **Object-Oriented programming language** developed by **James Gosling** in the early 1990s. Gosling wanted to alter and expand C++ however before long surrendered that for making another stage called **Green**. His team called their project “**Greentalk**” and its file extension was **.gt** and later became “**OAK**”. The name **Oak** was used by **Gosling** after an **oak tree** that was outside his office. Also, Oak is an image of solidarity and strength.

They later renamed it as “**JAVA**” as it was already a trademark by **Oak Technologies**. The following are the versions of java-

1. JDK Alpha and Beta (in 1995)
2. JDK 1.0 (23rd January, 1996)
3. JDK 1.1 (19th February, 1997)
- 4.J2SE 1.2 (8th December, 1998)
- 5.J2SE 1.3 (8th May, 2000)
- 6.J2SE 1.4 (6th February, 2002)
7. J2SE 5.0 (30th September, 2004)
- 8.Java SE 6 (11th December, 2006)
- 9.Java SE 7 (28th July, 2011)
- 10.Java SE 8 (18th March, 2014)
- 11.Java SE 9 (21st September, 2017)
- 12.Java SE 10 (20th March, 2018)
- 13.Java SE 11 (September, 2018)
- 14.Java SE 12 (March, 2019)
- 15.Java SE 13 (September, 2019)
- 16.Java SE 14 (March, 2020)
- 17.Java SE 15 (September, 2020)
- 18.Java SE 16 (March, 2021)
- 19.Java SE 17 (September, 2021)

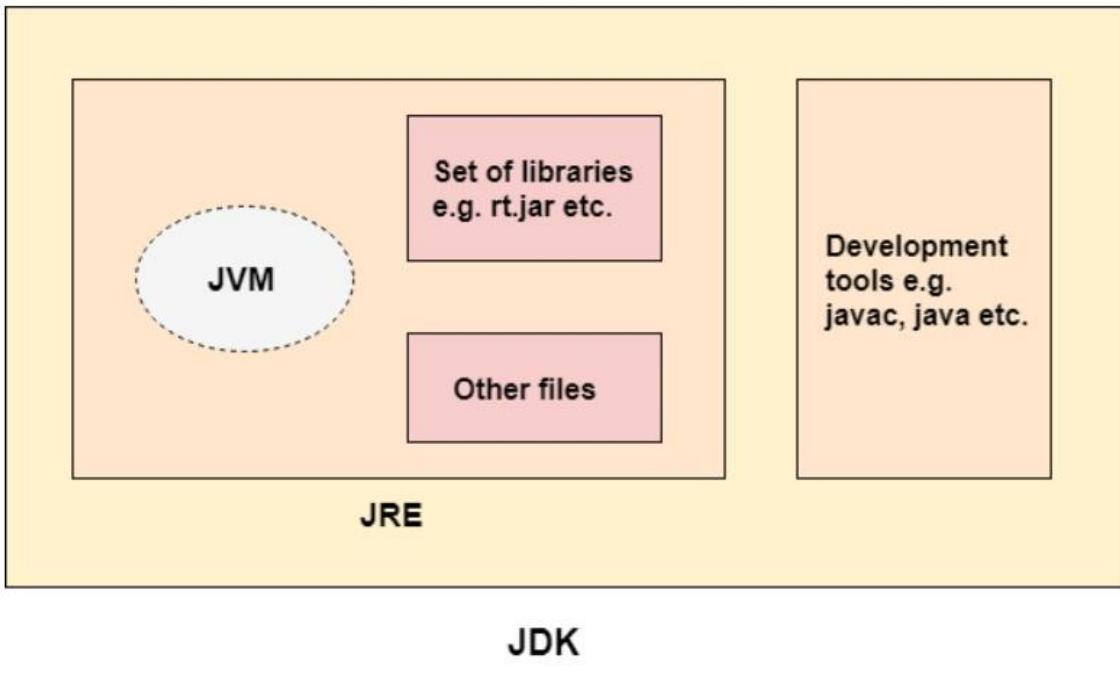
20.Java SE 18 (March, 2022)

Q.2 Explain JDK, JRE, JVM.

JDK: Java Development Kit implements the Java Language Specification and the Java Virtual Machine Specification and provides the Standard Edition of the Java Application Programming Interface. It exists physically for use.

JRE: Java runtime environment is a bunch of programming instruments which are utilized for creating java applications. It is the proper implementation of JVM. It contains a bunch of libraries + different records that JVM utilizes at runtime.

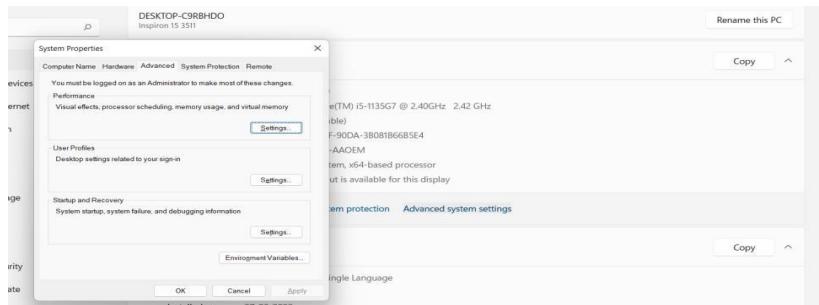
JVM: Java virtual machine is an abstract machine, provides a runtime environment for execution of bytecode. It gets loaded but not seen while loading or execution. It gets installed while installing JDK.



Q.3 How to set path of Environment Variables?

The path of environment variables needs to be set in java to use directories properly, without it javac, java etc cannot be used properly.

Step 1: Go to This PC → Advanced system settings



Step 2: Click on new path

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
path	C:\Program Files\Java\jdk-17.0.3.1\bin
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.MSC
PROCESSOR_ARCHITECTU...	AMD64
PROCESSOR_IDENTIFIER	Intel(R) Core(TM) i5-1135G7 CPU @ 2.40GHz

Step 3: Copy path till bin and give value



Step 4: Execute java and javac commands.

```
javac [options] <source files>
where possible options include:
  @<filename>                                Read options and filenames from file
  -Akey[=value]                                  Options to pass to annotation processors
  -add-modules <module>(<module>)*
    Root modules to resolve in addition to the initial modules, or all modules
    on the module path if <module> is ALL-MODULE-PATH.
  -boot-class-path <path>, --bootclasspath <path>
    Override location of bootstrap class files
  -class-path <path>, --classpath <path>, -cp <path>
    Specify where to find user class files and annotation processors
  -d <directory>                                 Specify where to place generated class files
  -deprecation
    Output source locations where deprecated APIs are used
  -encoding <encoding>                           Specify character encoding used by source files
  -endorseddirs <dirs>                           Override location of endorsed standards path
  -extdirs <dirs>                                 Override location of installed extensions
  -g                                     Generate all debugging info
  -g:(lines,vars,source)                         Generate only some debugging info
  -g:none                                    Generate no debugging info
  -h <directory>
    Specify where to place generated native header files
  -help, --help, -?
    Print this help message
  -help-extra, -X
    Print help on extra options
  -implicit:{(none,}class}
```

```
C:\Users\Vamika> java
Usage: java [options] <mainclass> [args...]
        (to execute a class)
    or  java [options] -jar <jarfile> [args...]
        (to execute a jar file)
    or  java [options] -m <module>/<mainclass> [args...]
    java [options] --module <module>/<mainclass> [args...]
        (to execute the main class in a module)
    or  java [options] <sourcefile> [args]
        (to execute a single source-file program)

Arguments following the main class, source file, -jar <jarfile>,
-m or --module <module>/<mainclass> are passed as the arguments to
main class.

where options include:

-cp <class search path of directories and zip/jar files>
-classpath <class search path of directories and zip/jar files>
--class-path <class search path of directories and zip/jar files>
    A ; separated list of directories, JAR archives,
    and ZIP archives to search for class files.
-p <module path>
--module-path <module path>...
    A ; separated list of directories, each directory
    is a directory of modules.
```

Q.4 Write a sample Hello world program and execute it on command line prompt and Linux Terminal.

Command Line Prompt



```

Command Prompt
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vamika> cd C:\java programs

C:\java programs> javac Sample.java

C:\java programs> java Sample
Hello World

```

Linux Terminal.

The screenshot shows a desktop environment with a dock containing icons for various applications like a file manager, browser, and system tools. A terminal window is open, displaying the following Java code:

```

class Sample
{
    public static void main (String args[])
    {
        System.out.println("Hello World");
    }
}

```

Below the code, the terminal shows the output of the compilation and execution command:

```

vamika@ubuntu:~/Desktop$ javac -version
javac 1.8.0_292
vamika@ubuntu:~/Desktop$ cd Desktop
vamika@ubuntu:~/Desktop$ javac Sample.java
vamika@ubuntu:~/Desktop$ java Sample
Hello World
vamika@ubuntu:~/Desktop$ ss

```

Q. Platform Independence of Java.

The meaning of platform-independent is that java compiled code (byte code) can run on all machines irrespective of the operating system.

A program is written in human-readable language. It may contain words, phrases, etc which the machine does not understand. For the source code to be understood by the machine, it needs to be a machine-level language. Here comes the role of a compiler. The compiler converts the high-level language (human language) in a format understood by the machines. A compiler is a program that translates the source code for another program from a programming language into executable code.

The executable code may be a sequence of machine instructions that can be executed by the CPU directly, or it may be an intermediate representation that is interpreted by a virtual machine. This intermediate representation in Java is the **Java Byte Code**. **Q. Compiled and Interpreted program in java.**

Java is a platform-independent programming language. It does not work on the one-step compilation. It involves a two-step execution, first through a compiler which is independent of OS; and second, in a virtual machine (JVM) which is custom-built. Firstly the source ‘.java’ file is passed through the compiler, which then encodes the source code into a machine-independent encoding, known as Bytecode. The content of each class contained in the source file is stored in a separate ‘.class’ file. While converting the source code into the bytecode.

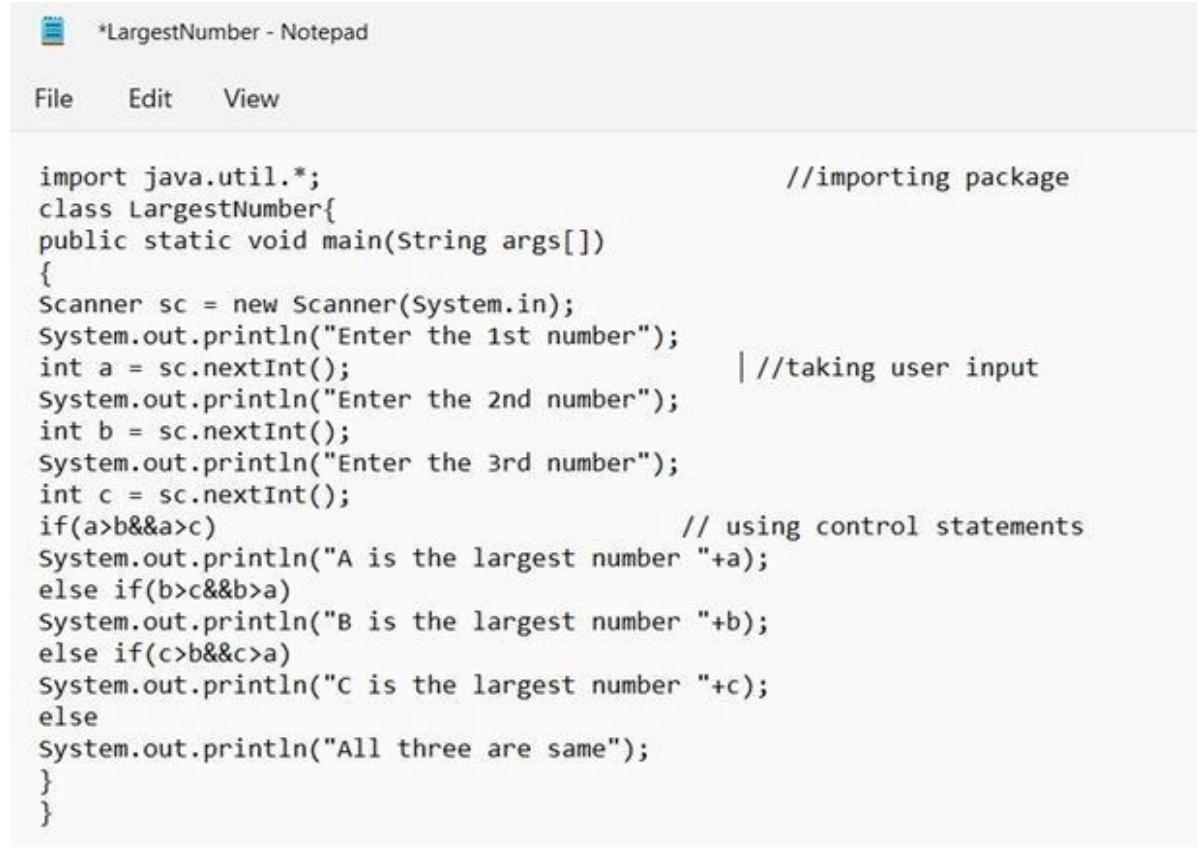
The class files generated by the compiler are independent of the machine or the OS, which allows them to be run on any system. To run, the main class file (the class that contains the method main) is passed to the JVM and then goes through three main stages before the final machine code is executed.

Compilation is done through **javac**.

Execution is done through **java**.

Experiment 2- Basics of Java

Q1. Write a program to find the largest of 3 given numbers.



```
*LargestNumber - Notepad
File Edit View

import java.util.*; //importing package
class LargestNumber{
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the 1st number");
int a = sc.nextInt(); //taking user input
System.out.println("Enter the 2nd number");
int b = sc.nextInt();
System.out.println("Enter the 3rd number");
int c = sc.nextInt();
if(a>b&&a>c) // using control statements
System.out.println("A is the largest number "+a);
else if(b>c&&b>a)
System.out.println("B is the largest number "+b);
else if(c>b&&c>a)
System.out.println("C is the largest number "+c);
else
System.out.println("All three are same");
}
}
```



```
Command Prompt
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vamika> cd C:\java programs

C:\java programs>javac LargestNumber.java

C:\java programs> java LargestNumber
Enter the 1st number
4
Enter the 2nd number
6
Enter the 3rd number
1
B is the largest number 6

C:\java programs>
```

```
C:\java programs> javac LargestNumber.java

C:\java programs> java LargestNumber
Enter the 1st number
4
Enter the 2nd number
4
Enter the 3rd number
4
All three are same

C:\java programs>
```

Q2. Write a program to implement a command line calculator to perform Addition, Subtraction, Multiplication, Division and Remainder.

```
public class Calculator {
    public static void main(String[] args)
    {
        int a,b;
        a=Integer.parseInt(args[0]);
        String c=args[1];           //taking character for operation
        b=Integer.parseInt(args[2]);   //conversion to integer
        int result=0;
        if(c.equals("+"))
        {
            result=a+b;
        }
        if(c.equals("-"))
        {
            result=a-b;
        }
        if(c.equals("X"))
        {
            result=a*b;
        }
        if(c.equals("/"))
        {
            result=a/b;
        }
        if(c.equals("%"))
        {
            result=a%b;
        }
        System.out.println("The result is = "+result);
    }
}
```

```

 Command Prompt
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vamika> cd C:\java programs

C:\java programs> javac Calculator.java

C:\java programs>java Calculator 4 + 5
The result is = 9

C:\java programs>_

C:\java programs> javac Calculator.java

C:\java programs> java Calculator 30 - 15
The result is = 15

C:\java programs>_

```

Q3. WAP to take today's date as user input from the command line and calculate the number of days since 1-January-2000.

 *Calender - Notepad

File Edit View

```

public class Calender{
public static void main(String args[])
{
int d=Integer.parseInt(args[0]);
int m=Integer.parseInt(args[1]);
int y=Integer.parseInt(args[2]);
int rd=0;
int rm=0;
int ry=0;
int a=0;
int mm,i;
for(i=2000; i<= y; i++)
{
if (i%4==0 && i%100!=0)
a=a+1; // leap year checking
}
if (y>=2000)
{
ry=(y-2000)*365; // remaining years
mm=(m-1);
for(i=1;i<=mm; i++)
{
if(i==1 || i==3 || i==5 || i==7 || i==8 || i==10 || i==12) // checking for months with 31 days
rm=rm+31;
else if (i==2)
rm=rm+28;
else
rm=rm+30;
}
rd=d-1;
System.out.println(ry+rm+rd+a);
}
else
{
System.out.println("Invalid date");
}
}

```

Command Prompt

```
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vamika> cd C:\java programs

C:\java programs> javac Calender.java

C:\java programs> java Calender 2 4 2003
1186
```

```
C:\java programs> javac Calender.java
```

```
C:\java programs>java Calender 21 6 2022
8206
```

```
C:\java programs>
```

Q4. WAP to print the following pattern using loops for 'n' rows.

```
import java.util.*;
public class Pattern
{
    public static void main(String args[])
    {
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter no of rows");
        int a= sc.nextInt();
        int i,j;
        for (i=1; i<=a; i++)           //outer loop
        {
            for( j=1; j<=i; j++)
                System.out.print("*");      //printing in single line
            System.out.println(" ");
        }
    }
}
```

 Command Prompt

```
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vamika> cd C:\java programs

C:\java programs> javac Pattern.java

C:\java programs> java Pattern
Enter no of rows
3
*
**
***
```

```
C:\java programs> javac Pattern.java

C:\java programs> java Pattern
Enter no of rows
5
*
**
***
****
*****
```

Q5. Write a program to accept 10 student's marks in an array, arrange it in ascending order, convert it into the following grades, Print marks and grades in the tabular form.

Less than 40: FAIL

Between 40 and 50: PASS

Between 51 and 75: MERIT

Greater than 75 and <= 100: DISTINCTION

```

import java.util.*;
public class Marks {

    public static void main(String[] args){
        int[] arr = new int[10];
        Scanner I= new Scanner(System.in);

        System.out.println("Enter Marks:");
        int i,swap,c,d;
        for( i = 0; i < 10; i++)
            arr[i] = I.nextInt();

            for ( c = 0; c < ( i - 1 ); c++)
                for ( d = 0; d < i - c - 1; d++)
                    if (arr[d] > arr[d+1])
                    {
                        swap = arr[d];
                        arr[d] = arr[d+1];
                        arr[d+1] = swap;
                    }

        System.out.println("\n\nPrinting Marks:");
        for( i = 0; i < 10; i++)
        {
            if(arr[i]<40)
            {
                System.out.println("\t|"+arr[i]+\t| " + " FAIL\t\t\t|");
            }
            else if(arr[i]>=41 && arr[i]<=60)
            {
                System.out.println("\t|"+arr[i]+\t| " + " MERIT\t\t\t|");
            }
            else if(arr[i]>=61 && arr[i]<=75)
            {
                System.out.println("\t|"+arr[i]+\t| " + " GOOD\t\t\t|");
            }
            else
            {
                System.out.println("\t|"+arr[i]+\t| " + " DISTINCTION\t\t|");
            }
        }
    }
}

```

C:\java programs> javac Marks.java

C:\java programs> java Marks
Enter Marks:
56 78 56 43 21 32 98 89 65 78

Printing Marks:

21	FAIL
32	FAIL
43	MERIT
56	MERIT
56	MERIT
65	GOOD
78	DISTINCTION
78	DISTINCTION
89	DISTINCTION
98	DISTINCTION

Experiment 3- Arrays

Q1. Write a program to accept three digits (0-9) and print all its possible combinations.

```
import java.util.*;
class Combination
{
public static void main( String args[])
{
Scanner sc= new Scanner(System.in);
System.out.println("Enter first digit");
int a= sc.nextInt();
System.out.println("Enter second digit");
int b= sc.nextInt();
System.out.println("Enter third digit");
int c= sc.nextInt();
int A[]={a,b,c};
int i,j,k;
for (i=0;i<3;i++)
for (j=0;j<3;j++)
for (k=0;k<3;k++)
if(i!=j && j!=k && k!=i)
System.out.println( (100*A[i]) + (10*A[j]) + A[k]);
}
}
```

Test Cases

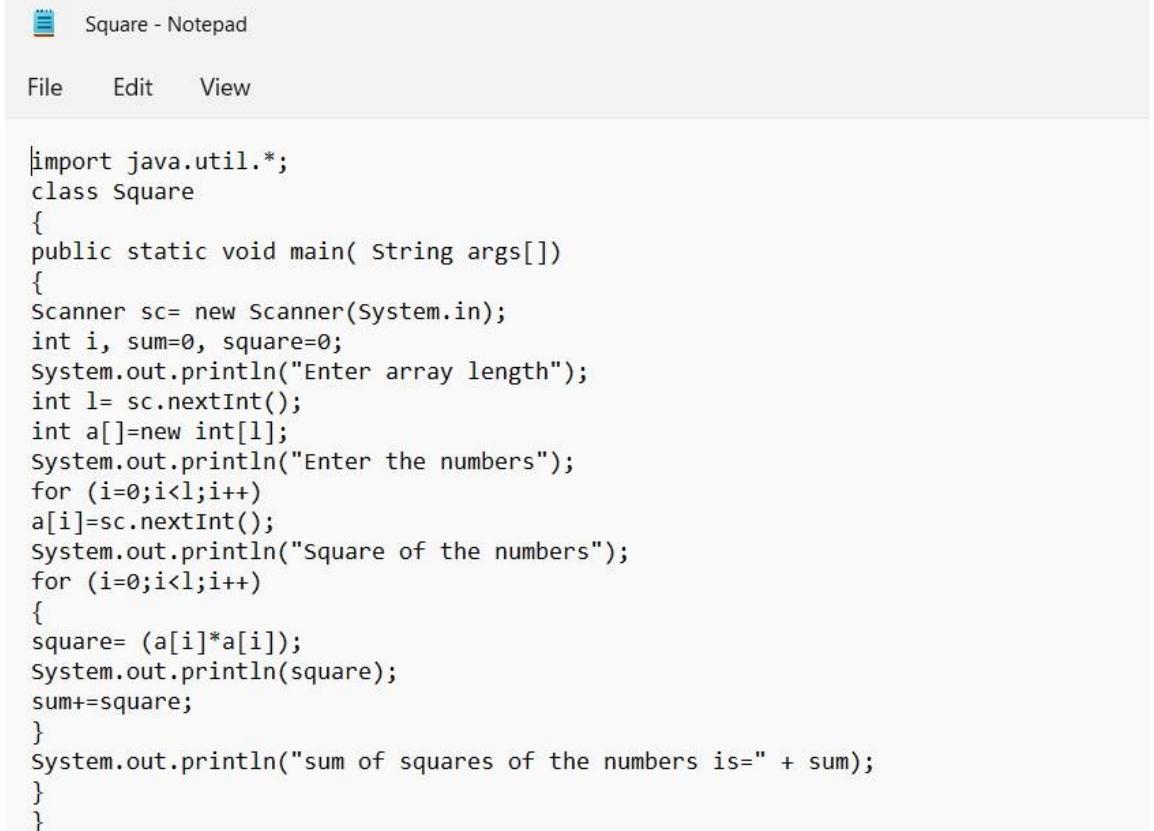
Input	Output
1,2,3	123,132,213,231,312,321
4,5,6	456,465,546,564,645,654
7,8,9	789,798,879,897,978,987

```
C:\java programs> javac Combination.java
C:\java programs> java Combination
Enter first digit
1
Enter second digit
2
Enter third digit
3
123
132
213
231
312
321
C:\java programs>
```

```
C:\java programs> java Combination
Enter first digit
4 5 6
Enter second digit
Enter third digit
456
465
546
564
645
654
```

```
C:\java programs> java Combination
Enter first digit
7
Enter second digit
8
Enter third digit
9
789
798
879
897
978
987
```

Q.2 Write a java program to accept 10 numbers in an array, and compute the square of each number. Print the sum of these numbers.



Square - Notepad

File Edit View

```
import java.util.*;
class Square
{
    public static void main( String args[])
    {
        Scanner sc= new Scanner(System.in);
        int i, sum=0, square=0;
        System.out.println("Enter array length");
        int l= sc.nextInt();
        int a[]=new int[l];
        System.out.println("Enter the numbers");
        for (i=0;i<l;i++)
            a[i]=sc.nextInt();
        System.out.println("Square of the numbers");
        for (i=0;i<l;i++)
        {
            square= (a[i]*a[i]);
            System.out.println(square);
            sum+=square;
        }
        System.out.println("sum of squares of the numbers is=" + sum);
    }
}
```

Test Cases

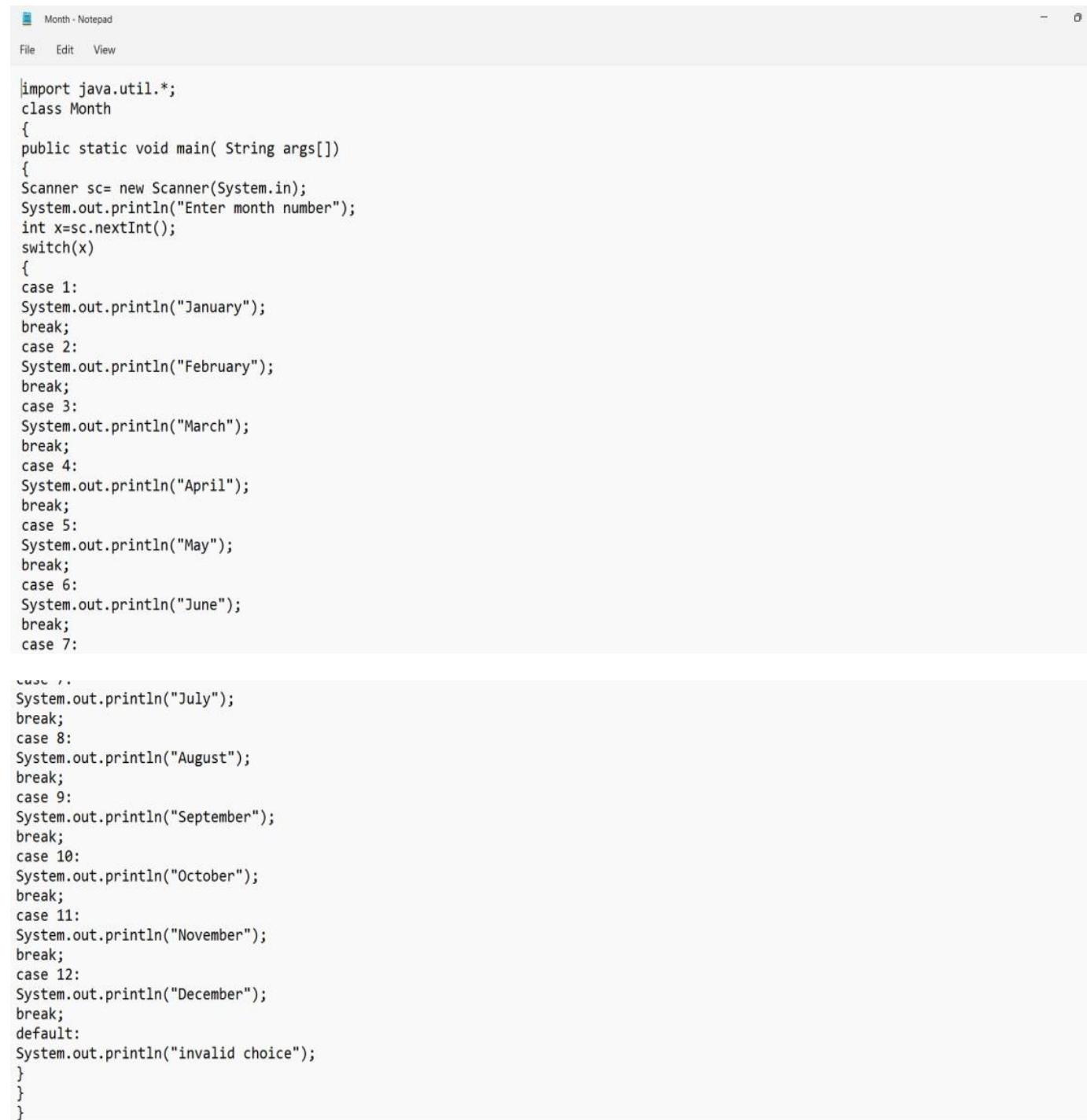
Input	Output
12,34,76,12,4,5,23,12,89,9	144,1156,5776,144,16,25,529,144,7921,81 Sum=15936
2,3,7,6	4,9,49,36 Sum=98
1,3,5,3,8,4	1,9,25,9,64,16 Sum=124

```
C:\java programs> java Square
Enter array length
10
Enter the numbers
12
34
76
12
4
5
23
12
89
9
Square of the numbers
144
1156
5776
144
16
25
529
144
7921
81
sum of squares of the numbers is=15936
```

```
C:\java programs> java Square
Enter array length
4
Enter the numbers
2
3
7
6
Square of the numbers
4
9
49
36
sum of squares of the numbers is=98
```

```
C:\java programs> java Square
Enter array length
6
Enter the numbers
1
3
5
3
8
4
Square of the numbers
1
9
25
9
64
16
sum of squares of the numbers is=124
```

Q.3 Write a program to input a number of a month (1-12) and print its equivalent name of the month.



```
Month - Notepad
File Edit View

import java.util.*;
class Month
{
public static void main( String args[])
{
Scanner sc= new Scanner(System.in);
System.out.println("Enter month number");
int x=sc.nextInt();
switch(x)
{
case 1:
System.out.println("January");
break;
case 2:
System.out.println("February");
break;
case 3:
System.out.println("March");
break;
case 4:
System.out.println("April");
break;
case 5:
System.out.println("May");
break;
case 6:
System.out.println("June");
break;
case 7:
System.out.println("July");
break;
case 8:
System.out.println("August");
break;
case 9:
System.out.println("September");
break;
case 10:
System.out.println("October");
break;
case 11:
System.out.println("November");
break;
case 12:
System.out.println("December");
break;
default:
System.out.println("invalid choice");
}
}
}
```

Test Cases

Input	Output
5	May
12	December
8	August

```
C:\java programs> javac Month.java
```

```
C:\java programs> java Month
Enter month number
5
May
```

```
C:\java programs>
```

```
C:\java programs> java Month
Enter month number
12
December
```

```
C:\java programs>
```

```
C:\java programs> java Month
Enter month number
8
August
```

```
C:\java programs>■
```

Q.4 Write a program to find the sum of all integers greater than 40 and less than 250 that are divisible by 5.



Sum - Notepad

File Edit View

```
import java.util.*;
class Sum
{
    public static void main( String args[])
    {
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter lower bound");
        int l= sc.nextInt();
        System.out.println("Enter upper bound");
        int u= sc.nextInt();
        System.out.println("Enter divisibility factor");
        int d= sc.nextInt();
        int i;
        int sum=0;
        for (i=l; i<u; i++)
        {
            if (i%d == 0)
                sum =sum + i;
        }
        System.out.println("Sum="+sum);
    }
}
```

Test Cases

Input	Output
0-20 5	30
20-40 2	290
70-100 3	855

```
C:\java programs> javac Sum.java

C:\java programs> java Sum
Enter lower bound
0
Enter upper bound
20
Enter divisibility factor
5
Sum=30

C:\java programs>
```

```
C:\java programs> java Sum
Enter lower bound
20
Enter upper bound
40
Enter divisibility factor
2
Sum=290

C:\java programs>
```

```
C:\java programs> java Sum
Enter lower bound
70
Enter upper bound
100
Enter divisibility factor
3
Sum=855

C:\java programs>
```

Experiment 4 - Inheritance

Q.1 Write a java program to show that private members of a super class cannot be accessed by a derived class. Also suggest a solution for it.



```
Child - Notepad

File Edit View

class Parent
{
private static int a=10;
}
class Child extends Parent
{
public static void main( String args[])
{
Child obj = new Child();
System.out.println(obj.a);
}
}
```



```
Select Command Prompt

C:\java programs> cd C:\java programs

C:\java programs> javac Child.java
Child.java:10: error: a has private access in Parent
System.out.println(obj.a);
                  ^
1 error

C:\java programs>_
```

SOLUTION:

Create a public void function in the private class to display private variable a and call it in the child class using an object of the Parent class as shown below.

```

class Parent
{
private static int a=10;
public void fun()
{
System.out.println("value of a =" + a);
}
}
class Child extends Parent
{
public static void main(String args[])
{
Parent obj = new Parent();
obj.fun();
}
}

```

Select Command Prompt

```

C:\java programs> javac Child.java

C:\java programs> java Child
value of a =10

C:\java programs>

```

Q.2 Write a java program to create a Player Class. Inherit the classes Cricket_Player, Football_Player & Hockey_Player from Player class. Include relevant member functions as required.

```

class Player{
    public String name = "Sachin Tendulkar";
    public String game = "Cricket";
}

class Cricket_player extends Player
{
    private static Player myPlayer = new Player();
    public void init()
    {
        name = "Sachin Tendulkar";
        game = "Cricket";
    }
    void display(){
        System.out.println("Name : "+myPlayer.name);
        System.out.println("Game : "+myPlayer.game);
    }
}

class Football_player extends Player{
    private static Player myPlayer = new Player();
    public void init() {
        myPlayer.name = "Peter Khan";
        myPlayer.game = "Football";
    }
    void display(){
        System.out.println("Name : "+myPlayer.name);
        System.out.println("Game : "+myPlayer.game);
    }
}

```

```

        System.out.println("Name : "+myPlayer.name);
        System.out.println("Game : "+myPlayer.game);
    }

class Hockey_player extends Player{
    private static Player myPlayer = new Player();
    public void init() {
        myPlayer.name = "Pratibha Singh";
        myPlayer.game = "Hockey";
    }
    void display(){
        System.out.println("Name : "+myPlayer.name);
        System.out.println("Game : "+myPlayer.game);
    }
}

public class Games {
    public static void main(String[] args) {
        Cricket_player cp = new Cricket_player();
        Football_player fp = new Football_player();
        Hockey_player hp = new Hockey_player();
        cp.init();
        fp.init();
        hp.init();

        cp.display();
        fp.display();
        hp.display();
    }
}

```

cmd Select Command Prompt

```

C:\java programs> javac Games.java

C:\java programs> java Games
Name : Sachin Tendulkar
Game : Cricket
Name : Peter Khan
Game : Football
Name : Pratibha Singh
Game : Hockey

```

Q.3 Write a java program to create a Worker class and derive the class DailyWorker and Salaried Worker from it. Every worker has a name and a salary rate. Write method ComPay (int hours) to compute the week pay of every worker. A Daily Worker is paid on the basis of the number of days he/she works. The Salaried Worker gets paid the wage for 40 hours a week no matter what the actual hours are. Test this program to calculate the pay of workers. You are expected to use the concept of polymorphism to write this program.

```

abstract class Worker
{
    public String name = "";
    public float perHourRate = 0;
    abstract void ComPay(int hours);
}
class DailyWorker extends Worker{
    DailyWorker(String n,float r){
        this.name = n;
        this.perHourRate = r;
    }
    void ComPay(int hours){
        float pay = this.perHourRate * hours;
        System.out.println("Pay For "+this.name+", is = Rs."+pay);
    }
}
class SalariedWorker extends Worker{
    SalariedWorker(String n,float r){
        this.name = n;
        this.perHourRate = r;
    }
    void ComPay(int hours){
        float pay = this.perHourRate * hours;
        System.out.println("Pay For "+this.name+", is = Rs."+pay);
    }
}

public class Work {
    public static void main(String[] args) {
        Worker suresh = new DailyWorker("Suresh Jha",150);
        Worker rajeev = new SalariedWorker("Rajeev Sharma", 40000);

        suresh.ComPay(40);
        rajeev.ComPay(40);
    }
}

```

```

C:\java programs> javac Work.java

C:\java programs> java Work
Pay For Suresh Jha, is = Rs.6000.0
Pay For Rajeev Sharma, is = Rs.1600000.0

C:\java programs>S_

```

Q.4 Consider the trunk calls of a telephone exchange. A trunk call can be ordinary, urgent, or lighting. The charges depend on the duration and the type of the call. Write a program using the concept of polymorphism in Java to calculate the charges.

```

class Trunkcalls{
    public static double rates_ordinary_seconds = 0.5;
    public static double rates_urgent_seconds = 1.5;
    public static double rates_lightning_seconds = 5;
    public double call(int timeInSeconds){
        double pay=0;
        pay = timeInSeconds*rates_ordinary_seconds;
        return pay;
    }
    public double call(int timeInSeconds,int flag){
        //urgent or lightning
        double pay=0;
        if(flag>0){
            pay = timeInSeconds*rates_urgent_seconds;
            //urgent
        }
        else{
            //lightning
            pay = timeInSeconds*rates_lightning_seconds;
        }
        return pay;
    }
}
public class Phone| {
    public static void main(String[] args) {
        Trunkcalls phone = new Trunkcalls();
        double payment = phone.call(100);
        System.out.println("Ordinary Payment = "+payment);

        payment = phone.call(100,1);
        System.out.println("Urgent Payment = "+payment);

        payment = phone.call(100,-1);
        System.out.println("Lightning Payment = "+payment);
    }
}

```

Command Prompt

```

Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vamika> cd C:\java programs

C:\java programs> javac Phone.java

C:\java programs> java Phone
Ordinary Payment = 50.0
Urgent Payment = 150.0
Lightning Payment = 500.0

C:\java programs>

```

Q.5 Design a class employee of an organization. An employee has a name, empid, and salary Write the default constructor, a constructor with parameters (name, empid, and salary) and methods to return name and salary. Also write a method increase Salary that raises the employee's salary by a certain user specified percentage. Derive a subclass Manager from employee. Add an instance variable named department to the manager class. Supply a test program that uses these classes and methods.

```

class Employee{
    public String name;
    private int empid;
    private int salary;
    Employee(){
        name = "";
        empid = 0;
        salary = 0;
    }
    Employee(String name, int empid, int salary){
        this.name = name;
        this.empid = empid;
        this.salary = salary;
    }
    String getName(){
        return this.name;
    }
    int getEmpid(){
        return this.empid;
    }
    int getSalary(){
        return this.salary;
    }
    void increaseSalary(double percent){
        this.salary+=this.salary*(percent/100);
    }
}
class Manager extends Employee{
    String department;
    Manager(){
        department = "Management";
    }
}
public class Emp {
    public static void main(String[] args) {
        Employee suresh = new Employee("Suresh Singh",1234,25000);
        System.out.println("Name : "+suresh.getName()+"\nEMPID : "+suresh.getEmpid()+"\nSalary : "+suresh.getSalary());
        suresh.increaseSalary(10);
        System.out.println("Increasing salary by 10%. New Salary = "+suresh.getSalary());
    }
}

```

Command Prompt

```

Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vamika> cd C:\java programs

C:\java programs> javac Emp.java

C:\java programs> java Emp
Name : Suresh Singh
EMPID : 1234
Salary : 25000
Increasing salary by 10%. New Salary = 27500

```

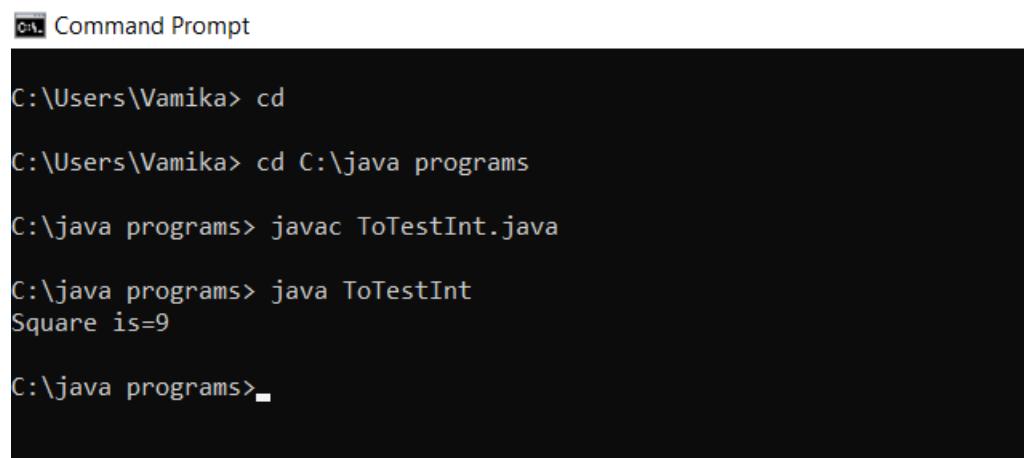
Experiment 5- Interfaces

Q1. a program to create an interface named test. In this interface the member function is square. Implement this interface in the arithmetic class. Create one new class called ToTestInt. In this class use the object of the arithmetic class.



```
ToTestInt - Notepad
File Edit View

interface Test
{
    public int square(int x);
}
class Arithmetic implements Test
{
    int a;
    public int square(int b)
    {
        a=b*b;
        return a;
    }
}
class ToTestInt
{
    public static void main(String args[])
    {
        Arithmetic obj=new Arithmetic();
        int z;
        z=obj.square(3);
        System.out.println("Square is=" +z);
    }
}
```



```
Command Prompt
C:\Users\Vamika> cd
C:\Users\Vamika> cd C:\java programs
C:\java programs> javac ToTestInt.java
C:\java programs> java ToTestInt
Square is=9
C:\java programs>
```

Q2. Write a program to create interface A, in this interface we have two methods meth1 and meth2. Implements this interface in another class named MyClass.

Ques2 - Notepad

File Edit View

```
interface A
{
    public int meth1(int a);
    public int meth2(int b);
}
class MyClass implements A
{
    int x,y;
    public int meth1(int p)
    {
        x=p*p*p;
        return x;
    }
    public int meth2(int q)
    {
        y=q*q;
        return y;
    }
}
class Ques2
{
    public static void main(String args[])
    {
        MyClass obj=new MyClass();
        int u,v;
        u=obj.meth1(2);
        v=obj.meth2(2);
        System.out.println("Cube is=" +u);
        System.out.println("Square is=" +v);
    }
}
```

```
C:\java programs> javac Ques2.java
C:\java programs> java Ques2
Cube is=8
Square is=4
C:\java programs>
```

Q3. Write a program in Java to show the usefulness of Interfaces as a place to keep a constant value of the program.

Ques3 - Notepad

File Edit View

```
interface B
{
final int a=1;
final int b=56;
final String c="Vamika";
public void Fun();
}
class A implements B
{
int sum=0;
public void Fun()
{
// error a=75;
sum=a+b;
System.out.println("Sum="+sum);
System.out.println("Name="+c);
}
}
class Ques3
{
public static void main(String args[])
{
A obj= new A();
obj.Fun();
}
}
```

```
C:\java programs> javac Ques3.java

C:\java programs>java Ques3
Sum=57
Name=Vamika

C:\java programs>
```

Q4. Write a program to create an Interface having two methods division and modulus. Create a class, which overrides these methods.

Ques4 - Notepad

File Edit View

```
interface A
{
    public int division(int a,int b);
    public int modulus(int x,int y);
}
class B implements A
{
    int p,q;
    public int division(int u,int v)
    {
        p=u/v;
        return p;
    }
    public int modulus(int t,int w)
    {
        q=t%w;
        return q;
    }
}
class Ques4
{
    public static void main(String args[])
    {
        B obj= new B();
        int g,h;
        g=obj.division(10,2);
        h=obj.modulus(10,2);
        System.out.println("Division="+g);
        System.out.println("Modulus="+h);
    }
}
```

```
C:\java programs>javac Ques4.java
C:\java programs>java Ques4
Division=5
Modulus=0
C:\java programs>
```

Q5. Write a program to create an interface StackInterface having methods push (), pop (), and display (). StackClass implements StackInterface. Class StackClass contains the main method which is having a switch case for selecting the particular operation of the stack.

```
import java.util.*;
interface StackInterface
{
    public void push();
    public void pop();
    public void display();
}
class StackClass implements StackInterface
{
    public void push()
    {
        System.out.println("push method");
    }
    public void pop()
    {
        System.out.println("pop method");
    }
    public void display()
    {
        System.out.println("display method");
    }

    public static void main(String args[])
    {
        StackClass obj=new StackClass();
        System.out.println("Select choice");
        System.out.println("1 for push");
        System.out.println("2 for pop");
        System.out.println("3 for display");
        System.out.println("Your choice");
        int f;
        Scanner sc= new Scanner(System.in);
        f= sc.nextInt();
        switch(f)
        {
            case 1:
                obj.push();
                break;
            case 2:
                obj.pop();
                break;
            case 3:
                obj.display();
                break;
            default:
                System.out.println("Invaidid choice");
        }
    }
}
```

```
C:\java programs>javac StackClass.java
```

```
C:\java programs> java StackClass
```

```
Select choice
```

```
1 for push
```

```
2 for pop
```

```
3 for display
```

```
Your choice
```

```
1
```

```
push method
```

```
C:\java programs>
```

```
C:\java programs> javac StackClass.java
```

```
C:\java programs>java StackClass
```

```
Select choice
```

```
1 for push
```

```
2 for pop
```

```
3 for display
```

```
Your choice
```

```
2
```

```
pop method
```

```
C:\java programs>javac StackClass.java
```

```
C:\java programs>java StackClass
```

```
Select choice
```

```
1 for push
```

```
2 for pop
```

```
3 for display
```

```
Your choice
```

```
3
```

```
display method
```

Experiment 6- Packages

Q1. Create the following packages pack1, pack2, and subpack. Package pack1 has public classes Addition and Subtraction, subpackage of pack1 is named subpack and has the public class Multiplication, and package pack2 has the Main class from where the different arithmetic operations implemented in the above packages are accessed.

The image shows three separate Notepad windows, each containing Java code for a specific class. The first window, titled 'Addition - Notepad', contains the code for the 'Addition' class. The second window, titled 'Subtraction - Notepad', contains the code for the 'Subtraction' class. The third window, titled 'Multiplication - Notepad', contains the code for the 'Multiplication' class. All three classes belong to the 'pack1' package, with 'Multiplication' being part of the 'subpack' subpackage.

```
package pack1;
public class Addition
{
    public int add()
    {
        return(10+20);
    }
}

package pack1;
public class Subtraction
{
    public int sub()
    {
        return(20-10);
    }
}

package pack1.subpack;
public class Multiplication
{
    public int mul()
    {
        return(10*20);
    }
}
```

Main - Notepad

File Edit View

```
package pack2;
import pack1.*;
import pack1.subpack.*;
public class Main
{
public static void main(String args[])
{
Addition a= new Addition();
Subtraction s= new Subtraction();
Multiplication m=new Multiplication();
System.out.println("Sum="+a.add());
System.out.println("Difference="+s.sub());
System.out.println("Product="+m.mul());
}
}
```

Name	Date modified	Type	Size
pack1	16-10-2022 17:05	File folder	
pack2	16-10-2022 17:05 size: 774 bytes	File folder	

Name	Date modified	Type	Size
subpack	16-10-2022 17:05	File folder	
Addition	16-10-2022 17:05	CLASS File	1 KB
Subtraction	16-10-2022 17:05	CLASS File	1 KB

> This PC > OS (C:) > java programs > pack2

Name	Date modified	Type	Size
Main	16-10-2022 17:05	CLASS File	2 KB

Type: CLASS File
Size: 1.09 KB
Date modified: 16-10-2022 17:05

Command Prompt

```
C:\java programs>javac -d . Addition.java Subtraction.java Multiplication.java Main.java
C:\java programs>java pack2.Main
Sum=30
Difference=10
Product=200
```

Q.2 Create the following packages pack1, pack2, and pack3. Package pack1 has the public classes Arithmetic and Relational. Package pack2 has the public classes Logical and Bitwise. These public classes perform the corresponding operations in relevant methods. Package pack3 has the Main class from where the implementations in the packages are accessed.

```
// Arithmetic
package pack1;
public class Arithmetic
{
    public int add(int x,int y)
    {
        return(x+y);
    }
    public int sub(int x,int y)
    {
        return(x-y);
    }
    public int mul(int x,int y)
    {
        return(x*y);
    }
    public int div(int x,int y)
    {
        return(x/y);
    }
    public int mod(int x,int y)
    {
        return(x%y);
    }
}
```

```
//Logical
package pack2;
public class Logical
{
    public boolean lor(int x,int y)
    {
        return((x>y)|| (y>x));
    }
    public boolean land(int x,int y)
    {
        return((x>y)&&(y>x));
    }
    public boolean lnot(int x, int y)
    {
        return(!(x==y));
    }
}
```

```
//Relational
package pack1;
public class Relational
{
    public void equalsto(int x, int y)
    {
        if(x==y)
            System.out.println(" True");
        else
            System.out.println(" False");
    }
    public void lessthan(int x, int y)
    {
        if(x<y)
            System.out.println(" True");
        else
            System.out.println(" False");
    }
    public void greaterthan(int x, int y)
    {
        if(x>y)
            System.out.println(" True");
        else
            System.out.println(" False");
    }
}
```

```

//Bitwise
package pack2;
public class Bitwise
{
public int band(int x,int y)
{
return(x&y);
}
public int bor(int x,int y)
{
return(x|y);
}
public int bxor(int x,int y)
{
return(x^y);
}
public int bcomp(int x)
{
return(~x);
}
public int srs(int x,int y)
{
return(x>>y);
}
public int rs(int x,int y)
{
return(x>>>y);
}
public int ls(int x,int y)
{
return(x<<y);
}
}

package pack3;
import pack1.*;
import pack2.*;
import java.util.*;
public class Main
{
public static void main(String args[])
{
Arithmetict h=new Arithmetict();
Relational r= new Relational();
Logical l= new Logical();
Bitwise p= new Bitwise();
Scanner sc=new Scanner(System.in);
System.out.println("Press 1 to start and 0 to exit");
System.out.println("enter your choice");
int c;
c= sc.nextInt();
do
{
System.out.println("enter 2 numbers") ;
int a, b;
a=sc.nextInt();
b=sc.nextInt();
System.out.println("Press 1 to add");
System.out.println("Press 2 to sub");
System.out.println("Press 3 to mul");
System.out.println("Press 4 to div");
System.out.println("Press 5 to mod");
System.out.println("Press 6 to equalsto");
System.out.println("Press 7 to lessthan");
System.out.println("Press 8 to greaterthan");
System.out.println("Press 9 to land");
System.out.println("Press 10 to lor");
System.out.println("Press 11 to lnot");
System.out.println("Press 12 to band");
System.out.println("Press 13 to bor");
System.out.println("Press 14 to bxor");
System.out.println("Press 15 to bcomp");
System.out.println("Press 16 to srs");
}
}

```

```
System.out.println("Press 1/ to rs");
System.out.println("Press 18 to ls");
System.out.println("enter your choice");
int x;
x=sc.nextInt();
switch(x)
{
case 1:
System.out.println("Addition="+h.add(a,b));
break;
case 2:
System.out.println("Subtraction="+h.sub(a,b));
break;
case 3:
System.out.println("Product="+h.mul(a, b));
break;
case 4:
System.out.println("Divison="+h.div(a,b));
break;
case 5:
System.out.println("Remainder="+h.mod(a,b));
break;
case 6:
r.equalsto(a,b);
break;
case 7:
r.lessthan(a,b);
break;
case 8:
r.greaterthan(a,b);
break;
case 9:
System.out.println(" logical And="+l.land(a,b));
break;
case 10:
System.out.println(" logical or="+l.lor(a, b));
break;
case 11:
System.out.println(" logical not="+l.lnot(a,b));
break;
---- 12 ----
```

```

        break;
    case 11:
        System.out.println(" logical not="+l.lnot(a,b));
        break;
    case 12:
        System.out.println(" bitwise And="+p.band(a, b));
        break;
    case 13:
        System.out.println(" bitwise or="+p.bor(a, b));
        break;
    case 14:
        System.out.println(" bitwise xor="+p.bxor(a, b));
        break;
    case 15:
        System.out.println(" Choose a or b" );
        char o;
        o=sc.next().charAt(0);
        if(o=='a')
            System.out.println(" bitwise comp="+p.bcomp(a));
        else if(o=='b')
            System.out.println(" bitwise comp="+p.bcomp(b));
        else
            System.out.println("invalid choice");
        break;
    case 16:
        System.out.println(" signed right shift="+p.srs(a, b));
        break;
    case 17:
        System.out.println(" unsigned right shift="+p.rs(a,b));
        break;
    case 18:
        System.out.println(" bitwise left shift="+p.ls(a, b));
        break;
    default:
        System.out.println("invalid choice");
    }
    System.out.println("enter your choice");
    c= sc.nextInt();
}while(c!=0);
}
}

```

```
C:\java programs\lab6> javac -d . Arithmetic.java Relational.java Logical.java B  
C:\java programs\lab6> java pack3.Main  
Press 1 to start and 0 to exit  
enter your choice  
1  
enter 2 numbers  
10 20  
Press 1 to add  
Press 2 to sub  
Press 3 to mul  
Press 4 to div  
Press 5 to mod  
Press 6 to equalsto  
Press 7 to lessthan  
Press 8 to greaterthan  
Press 9 to land  
Press 10 to lor  
Press 11 to lnot  
Press 12 to band  
Press 13 to bor  
Press 14 to bxor  
Press 15 to bcomp  
Press 16 to srs  
Press 17 to rs  
Press 18 to ls  
enter your choice  
1  
Addition=30
```

```
enter your choice  
1  
enter 2 numbers  
30 20  
Press 1 to add  
Press 2 to sub  
Press 3 to mul  
Press 4 to div  
Press 5 to mod  
Press 6 to equalsto  
Press 7 to lessthan  
Press 8 to greaterthan  
Press 9 to land  
Press 10 to lor  
Press 11 to lnot  
Press 12 to band  
Press 13 to bor  
Press 14 to bxor  
Press 15 to bcomp  
Press 16 to srs  
Press 17 to rs  
Press 18 to ls  
enter your choice  
2  
Subtraction=10  
enter your choice
```

```
enter 2 numbers
10 2
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
3
Product=20
enter your choice
1
enter 2 numbers
40 6
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
4
Divison=6
enter your choice
```

Command Prompt - java pack3.Main

```
enter 2 numbers
58 3
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
5
Remainder=1
enter your choice
1
enter 2 numbers
3 4
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
6
False
```

```
enter 2 numbers
6 8
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
7
    True
enter your choice
1
enter 2 numbers
8 3
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
8
    True
```

```
enter 2 numbers
0 3
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
9
    logical And=false
enter your choice
```

```
enter 2 numbers
0 0
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
10
 logical or=false
```

```
enter 2 numbers
2 4
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
11
 logical not=true
enter your choice
```

```
enter 2 numbers
5 7
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
13
 bitwise or=7
enter your choice
```

```
enter 2 numbers
9
4
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
14
bitwise xor=13
enter your choice
```

```
enter 2 numbers
4 3
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
15
Choose a or b
a
bitwise comp=-5
enter your choice
```

```
enter 2 numbers
16
2
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
16
signed right shift=4
```

```
enter 2 numbers
4 1
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
17
| unsigned right shift=2
enter your choice
```

```
enter 2 numbers
9 2
Press 1 to add
Press 2 to sub
Press 3 to mul
Press 4 to div
Press 5 to mod
Press 6 to equalsto
Press 7 to lessthan
Press 8 to greaterthan
Press 9 to land
Press 10 to lor
Press 11 to lnot
Press 12 to band
Press 13 to bor
Press 14 to bxor
Press 15 to bcomp
Press 16 to srs
Press 17 to rs
Press 18 to ls
enter your choice
18
| bitwise left shift=36
enter your choice
```

Experiment 7 - Exceptions

Q1. Write a program to implement scenarios to handle the following built-in exceptions: (1) **ArithmaticException**, (2) **NullPointerException**, (3) **ArrayIndexOutOfBoundsException**, (4) **StringIndexOutOfBoundsException**, (5) **ClassNotFoundException**, (6) **IllegalArgumentException**, (7) **NumberFormatException**, (8) **CloneNotSupportedException**, (9) **InputMismatchException**. The program is menu-driven with an option to exit the application as required.

```
import java.util.*;
class MyException extends Exception
{
private String s;
public MyException(String s)
{
    super(s);
    this.s=s;
}
public String toString()
{
return s;
}
protected Object clone()
throws CloneNotSupportedException
{
return super.clone();
}
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
System.out.println("1 to enter and 0 to continue");
int c,f;
c=sc.nextInt();
while(c!=0)
{
System.out.println("Press 1 for Arithmatic exception");
System.out.println("Press 2 for Number Format exception");
System.out.println("Press 3 for Null pointer exception");
System.out.println("Press 4 for Array index out of bound exception");
System.out.println("Press 5 for String index out of bound exception");
System.out.println("Press 7 for illegal argument exception");
System.out.println("Press 8 for Clone not supported");
System.out.println("Press 9 for Input mismatch");
System.out.println("your choice");
f=sc.nextInt();
switch(f)
{
case 1:
```

```
case 1:  
try  
{  
int div=50/0;  
}  
catch(ArithmetricException e)  
{  
e.printStackTrace();  
}  
break;  
case 2:  
try  
{  
String a="abcd";  
int x=Integer.parseInt(a);  
}  
catch(NumberFormatException e)  
{  
e.printStackTrace();  
}  
break;  
case 3:  
try  
{  
String a=null;  
int x=a.length();  
}  
catch(NullPointerException e)  
{  
e.printStackTrace();  
}  
break;  
case 4:  
try  
{  
int a[]={1,2,3,4};  
a[7]=10;  
}
```

```
a[7]=10;
}
catch(ArrayIndexOutOfBoundsException e)
{
e.printStackTrace();
}
break;
case 5:
try
{
String v= "Vamika";
char a= v.charAt(9);
}
catch(StringIndexOutOfBoundsException e)
{
e.printStackTrace();
}
break;
case 6:
try
{
Class.forName("The Class do not Exist");
}
catch(ClassNotFoundException e)
{
e.printStackTrace();
}
break;
case 7:
System.out.println("enter two numbers for division");
int a,b,d;
a=sc.nextInt();
b=sc.nextInt();
if (b==0)
throw new IllegalArgumentException("B not zero");
else
d=a/b;
break;
case 8:
MyException ob= new MyException("vamika");
```

```

try
{
MyException obj=(MyException)ob.clone();
System.out.println(ob);
}
catch (CloneNotSupportedException e)
{
    e.printStackTrace();
}
break;
case 9:
try
{
System.out.println("enter an integer");
int h;
h=sc.nextInt();
}
catch (InputMismatchException e)
{
e.printStackTrace();
}
break;
default:
System.out.println("invalid choice");
}
System.out.println("press 0 to exit");
c=sc.nextInt();
}
}
}
}

```

```

C:\java programs>java MyException
1 to enter and 0 to continue
1
Press 1 for Arithmetic exception
Press 2 for Number Format exception
Press 3 for Null pointer exception
Press 4 for Array index out of bound exception
Press 5 for String index out of bound exception
Press 7 for illegal argument exception
Press 8 for Clone not supported
Press 9 for Input mismatch
your choice
1
java.lang.ArithmaticException: / by zero
    at MyException.main(MyException.java:42)
press 0 to exit

```

```
Press 1 for Arithmetic exception
Press 2 for Number Format exception
Press 3 for Null pointer exception
Press 4 for Array index out of bound exception
Press 5 for String index out of bound exception
Press 7 for illegal argument exception
Press 8 for Clone not supported
Press 9 for Input mismatch
your choice
2
java.lang.NumberFormatException: For input string: "abcd"
    at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
    at java.base/java.lang.Integer.parseInt(Integer.java:668)
    at java.base/java.lang.Integer.parseInt(Integer.java:786)
    at MyException.main(MyException.java:53)
press 0 to exit
```

```
Press 1 for Arithmetic exception
Press 2 for Number Format exception
Press 3 for Null pointer exception
Press 4 for Array index out of bound exception
Press 5 for String index out of bound exception
Press 7 for illegal argument exception
Press 8 for Clone not supported
Press 9 for Input mismatch
your choice
3
java.lang.NullPointerException: Cannot invoke "String.length()" because "<local4>" is null
    at MyException.main(MyException.java:64)
press 0 to exit
```

```
Press 1 for Arithmetic exception
Press 2 for Number Format exception
Press 3 for Null pointer exception
Press 4 for Array index out of bound exception
Press 5 for String index out of bound exception
Press 7 for illegal argument exception
Press 8 for Clone not supported
Press 9 for Input mismatch
your choice
4
java.lang.ArrayIndexOutOfBoundsException: Index 7 out of bounds for length 4
    at MyException.main(MyException.java:75)
press 0 to exit
```

```
Press 1 for Arithmetic exception
Press 2 for Number Format exception
Press 3 for Null pointer exception
Press 4 for Array index out of bound exception
Press 5 for String index out of bound exception
Press 7 for illegal argument exception
Press 8 for Clone not supported
Press 9 for Input mismatch
your choice
5
java.lang.StringIndexOutOfBoundsException: String index out of range: 9
    at java.base/java.lang.StringLatin1.charAt(StringLatin1.java:48)
    at java.base/java.lang.String.charAt(String.java:1515)
    at MyException.main(MyException.java:86)
press 0 to exit
```

```
Press 3 for Null pointer exception
Press 4 for Array index out of bound exception
Press 5 for String index out of bound exception
Press 7 for illegal argument exception
Press 8 for Clone not supported
Press 9 for Input mismatch
your choice
6
java.lang.ClassNotFoundException: The Class do not Exist
    at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.java:641)
    at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:188)
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:520)
    at java.base/java.lang.Class.forName0(Native Method)
    at java.base/java.lang.Class.forName(Class.java:375)
    at MyException.main(MyException.java:96)
press 0 to exit
```

```
Press 1 for Arithmetic exception
Press 2 for Number Format exception
Press 3 for Null pointer exception
Press 4 for Array index out of bound exception
Press 5 for String index out of bound exception
Press 7 for illegal argument exception
Press 8 for Clone not supported
Press 9 for Input mismatch
your choice
7
enter two numbers for division
3 0
Exception in thread "main" java.lang.IllegalArgumentException: B not zero
    at MyException.main(MyException.java:109)
```

```
Press 1 for Arithmetic exception
Press 2 for Number Format exception
Press 3 for Null pointer exception
Press 4 for Array index out of bound exception
Press 5 for String index out of bound exception
Press 7 for illegal argument exception
Press 8 for Clone not supported
Press 9 for Input mismatch
your choice
8
java.lang.CloneNotSupportedException: MyException
    at java.base/java.lang.Object.clone(Native Method)
    at MyException.clone(MyException.java:17)
    at MyException.main(MyException.java:117)
press 0 to exit
```

```
Press 5 for String index out of bound exception
Press 7 for illegal argument exception
Press 8 for Clone not supported
Press 9 for Input mismatch
your choice
9
enter an integer
fg
java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:939)
    at java.base/java.util.Scanner.next(Scanner.java:1594)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
    at MyException.main(MyException.java:130)
press 0 to exit
```

Q.2 Design a banking app that has an ‘Account’ class to represent a bank account where a customer can deposit and withdraw money. The app should not permit the customer to withdraw money that exceeds the bank balance. Create a custom exception called “NoSufficientFundException” to handle this scenario and show a more meaningful message to its customers.

```
import java.util.Scanner;
class NoSufficientFundException extends Exception
{
public NoSufficientFundException(String s)
{
super(s);
}
}
public class Bank
{

String accno;
String name;
String acc_type;
double balance;
Scanner sc = new Scanner(System.in);

public void openAccount()
{
    System.out.println("Enter Account No: ");
    accno = sc.next();
    System.out.println("Enter Account type: ");
    acc_type = sc.next();
    System.out.println("Enter Name: ");
    name = sc.next();
    System.out.println("Enter Balance: ");
    balance = sc.nextDouble();
}

public void showAccount()
{
    System.out.println("Name of account holder: " + name);
    System.out.println("Account no.: " + accno);
    System.out.println("Account type: " + acc_type);
    System.out.println("Balance: " + balance);
}

public void deposit()
{
    double amt.
```

```

public void deposit()
{
    double amt;
    System.out.println("Enter the amount you want to deposit: ");
    amt = sc.nextDouble();
    balance = balance + amt;
    System.out.println("Balance after withdrawal: " + balance);
}

public void withdrawal()
{
    double amt;
    System.out.println("Enter the amount you want to withdraw: ");
    amt = sc.nextDouble();
    if (balance >= amt)
    {
        balance = balance - amt;
        System.out.println("Balance after withdrawal: " + balance);
    }
    else
    {
        try
        {
            System.out.println("Your balance is less than " + amt + "\tTransaction failed...!!!");
            throw new NoSufficientFundException("No Sufficient Funds");
        }
        catch(NoSufficientFundException e)
        {
            e.printStackTrace();
        }
    }
}

public static void main(String args[])
{
    Scanner sc=new Scanner(System.in);
    Bank b= new Bank();
    b.openAccount();
    . . .
}

public static void main(String args[])
{
    Scanner sc=new Scanner(System.in);
    Bank b= new Bank();
    b.openAccount();
    b.showAccount();
    System.out.println("Press 1 to deposit and 2 to withdraw");
    int c;
    c=sc.nextInt();
    if(c==1)
        b.deposit();
    else if(c==2)
        b.withdrawal();
    else
        System.out.println("enter valid choice");
}
}

```

```

C:\java programs> javac Bank.java

C:\java programs>java Bank
Enter Account No:
272927
Enter Account type:
Standard
Enter Name:
Vamika
Enter Balance:
378690.46
Name of account holder: Vamika
Account no.: 272927
Account type: Standard
Balance: 378690.46
Press 1 to deposit and 2 to withdraw
1
Enter the amount you want to deposit:
4749
Balance after withdrawal: 383439.46

```

```
C:\java programs> java Bank
Enter Account No:
899707
Enter Account type:
Regular
Enter Name:
Vamika
Enter Balance:
37892
Name of account holder: Vamika
Account no.: 899707
Account type: Regular
Balance: 37892.0
Press 1 to deposit and 2 to withdraw
2
Enter the amount you want to withdraw:
8966758
Your balance is less than 8966758.0      Transaction failed...!!
NoSufficientFundException: No Suffecient Funds
    at Bank.withdrawal(Bank.java:62)
    at Bank.main(Bank.java:84)
```

Q3. Write a program to demonstrate Exception Propagation.

```
class Simple
{
    int a,b;
    void m(int a, int b) {
        this.a=a;
        this.b=b;
        int data = a/b;
    }

    void n()
    {
        m(a,b);
    }

    void p()
    {
        try
        {
            n();
        }
        catch(ArithmetricException e)
        {
            e.printStackTrace();
        }
    }

    public static void main(String args[])
    {
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter two numbers");
        int a,b;
        a=sc.nextInt();
        b=sc.nextInt();
        Simple obj = new Simple();
        obj.p();
    }
}
```

```
C:\java programs>javac Simple.java

C:\java programs> java Simple
Enter two numbers
3 0
java.lang.ArithmetricException: / by zero
    at Simple.m(Simple.java:10)
    at Simple.n(Simple.java:15)
    at Simple.p(Simple.java:22)
    at Simple.main(Simple.java:38)
```

Experiment 8- Strings

Q1. Write a program to find the first and last occurrence of a character or substring in a given string.

```
import java.util.*;
class String1
{
    public static void main(String args[])
    {
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter the string");
        String str= sc.nextLine();
        int n = str.length();
        System.out.println("Enter the string/ character to be searched");
        String search= sc.nextLine();

        int first = str.indexOf(search);
        int last = str.lastIndexOf(search);

        System.out.println("First: " + first);
        System.out.println("Last: " + last);
    }
}
```

Windows Command Prompt
Microsoft Windows [Version 10.0.22000.1098]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vamika> cd C:\java programs

C:\java programs> java String1
Enter the string
vamika
Enter the string/ character to be searched
a
First: 1
Last: 5

C:\java programs> java String1
Enter the string
vanhcovahfdva
Enter the string/ character to be searched
va
First: 0
Last: 10

Q2. Write a program that converts all characters of a string into capital letters. (Use StringBuffer to store a string). Don't use built-in functions.

```
import java.util.*;
public class String2
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a String");
        String str=sc.nextLine();
        int i=0;
        String s="";
        StringBuffer st = new StringBuffer(s);
        for(i=0;i<str.length();i++)
        {
            char ch=str.charAt(i);
            if(ch==' ')
            {
                st.append(" ");
            }
            int a=(int)ch;
            int b = a - 32;
            ch=(char)b;
            st.append(ch);
        }
        System.out.println(st);
    }
}
```

cmd Select Command Prompt

```
C:\java programs>java String2
Enter a String
vamika is the best
VAMIKA IS THE BEST
```

```
C:\java programs>■
```

Q.3 Write a program in Java to read a statement from the console, convert it into upper case, and again print it on the console. (Don't use built-in functions)

```
import java.util.*;
class String3
{
    public String Upper_Case(String str)
    {
        String s="";
        int i=0;
        for(i=0;i<str.length();i++)
        {
            char ch=str.charAt(i);
            if(ch==' ')
            {
                s=s+" ";
            }
            int a=(int)ch;
            int b = a - 32;
            ch=(char)b;
            s=s+ch;
        }
        return s;
    }

    public static void main(String args[])
    {
        Scanner sc =new Scanner(System.in);
        System.out.println("Enter a String");
        String str=sc.nextLine();
        String3 ob = new String3();
        String s=ob.Upper_Case(str);
        System.out.println(s);
    }
}
```

```
C:\java programs>java String3
Enter a String
vamika mahajan
The Upper Case String is VAMIKA MAHAJAN
```

```
C:\java programs>A
```

Q.4 Write a program in Java to create a String object. Initialize this object with your name. Find the length of your name using the appropriate String method. Find whether the character ‘a’ is in your name or not; if yes, find the number of times ‘a’ appears in your name. Print locations of occurrences of ‘a’. Try the same for different String objects.

```
import java.util.*;
public class String4
{
    public static void main(String[] args)
    {
        String str="vamika";
        int l= str.length();

        System.out.println("Length of string: "+l);
        int count=str.indexOf('a');

        if(count==-1)

            System.out.println("a is not present!");

        else

        {

            for(int i=0;i<l;i++)
            {
                char c=str.charAt(i);

                if('a'==c)

                    System.out.println("a is present in string at index: "+(i+1));
            }
        }
    }
}
```

```
C:\java programs> java String4
Length of string: 6
a is present in string at index: 1
a is present in string at index: 5
```

Q5. Write a Java code that converts int to Integer, Integer to String, String to int, int to String, String to Integer, and Integer to int.

```
import java.util.*;
import java.lang.Integer;

public class Wrapper1
{
    public static void main(String args[]){
        Scanner in = new Scanner(System.in);
        System.out.print("Enter an int value : ");
        int i = in.nextInt();
        Integer obj = new Integer(i);
        System.out.println("Convert int to Integer : "+obj);
        String str = obj.toString();
        System.out.println("Convert Integer to String : " + str);
        int num = Integer.parseInt(str);
        System.out.println("Convert String to int : "+num);
        String st = Integer.toString(obj);
        System.out.println("Convert int to String : "+st);
        Integer k = Integer.valueOf(st);
        System.out.println("Convert String to Integer : "+k);
        int l = obj.intValue();
        System.out.println("Convert Integer to int : "+l);
    }
}
```

```
C:\java programs> java Wrapper1
Enter an int value : 6
Convert int to Integer : 6
Convert Integer to String : 6
Convert String to int : 6
Convert int to String : 6
Convert String to Integer : 6
Convert Integer to int : 6

C:\java programs>
```

Q6. Write a Java code that converts a float to Float, Float to String, String to float, float to String, String to Float,

```
import java.util.*;
import java.lang.Float;

public class Wrapper2 {
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the float value : ");
        float a = sc.nextFloat();
        Float obj = new Float(a);
        System.out.println("Convert float to Float : "+obj);
        String s = obj.toString();
        System.out.println("Convert Float to String : " + s);
        float num = Float.parseFloat(s);
        System.out.println("Convert String to float : "+num);
        String st =String.valueOf(a);
        System.out.println("Convert float to String : "+st);
        float k = Float.valueOf(st);
        System.out.println("Convert String to Float : "+k);
        float l = obj.floatValue();
        System.out.println("Convert Float to float : "+l);
    }
}
```

```
C:\java programs> java Wrapper2
Enter the float value : 5.9
Convert float to Float : 5.9
Convert Float to String : 5.9
Convert String to float : 5.9
Convert float to String : 5.9
Convert String to Float : 5.9
Convert Float to float : 5.9
```

Experiment 9- Threads

Q 1- Write a program for generating 3 threads. Thread-1 and Thread-2 are instances of class Demo1 which inherits the Thread class. Thread-3 is an instance of class Demo2 which implements the Runnable interface. Thread-1 prints even numbers from 1 to n. Thread-2 prints odd numbers from 1 to m. Thread-3 prints Fibonacci series up to y. Implement the program using synchronization methods of Thread synchronization.

```
| class Table
{
    synchronized void peven(int n)
    {
        System.out.println("Even Numbers\n");
        for(int i=1;i<=n;i++)
        {
            if(i%2==0)
                System.out.println(i);
        }
    }
    synchronized void podd(int m)
    {
        System.out.println("Odd Numbers\n");
        for(int i=1;i<=m;i++)
        {
            if(i%2==1)
                System.out.println(i);
        }
    }
    synchronized void fibo(int y)
    {
        int n1=0,n2=1,n3;
        System.out.println("Fibonacci series\n");
        System.out.print(n1 + " " + n2);
        for(int i=2;i<y;i++)
        {
            n3=n1+n2;
            System.out.print(" " + n3);
            n1=n2;
            n2=n3;
        }
        System.out.println("\n");
    }
}
class Demo extends Thread
{
```

```

class Demo extends Thread
{
    Table t;
    Demo(Table obj)
        { t=obj; }
    public void run()
    {
        if(Thread.currentThread().getName()=="T1")
            t.peven(20);
        else
            t.podd(20);
    }
}

class Demo1 implements Runnable
{
    Table t;
    Demo1(Table obj)
        { t = obj; }
    public void run()
    {
        t.fibo(20);
    }
}

public class thread1
{
    public static void main(String args[])
    {
        Table t = new Table();
        Demo t1= new Demo(t);
        Demo t2 = new Demo(t);
        t1.setName("T1");
        Demo1 d = new Demo1(t);
        Thread t3 = new Thread(d);
        t1.start();
        t2.start();
        t3.start();
    }
}

```

cmd Command Prompt

```
C:\java programs>java thread1
Even Numbers

2
4
6
8
10
12
14
16
18
20
Fibonacci series

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181

Odd Numbers

1
3
5
7
9
11
13
15
17
19

C:\java programs>
```

Q2- Design a java program with class "Travel" with a data member available seats and member functions book(int) and cancel(int). The object of class Travel is shared with 2 threads and each thread can perform either booking or cancellation of tickets. Implement the program using Thread cooperation.

```

class Travel
{
    int available_seats = 150;
    synchronized void book(int seats)
    {
        if(seats > available_seats)
        {
            try{
                wait();
            }
            catch(InterruptedException ex){}
        }
        available_seats = available_seats - seats;
        System.out.println("Seats booked successfully!!");
    }
    synchronized void cancel(int seats)
    {
        available_seats = available_seats + seats;
        System.out.println("Seats canceled successfully!!");
        notify();
    }
}

public class thread2
{
    public static void main(String args[])
    {
        Travel t= new Travel();
        Thread t1=new Thread()
        {
            public void run() {
                t.book(120);  }
        };
        t1.start();
        Thread t2=new Thread()
        {
            public void run() {
                t.cancel(12);  }
        };
        t2.start();
    }
}

```

Command Prompt

```
C:\java programs>javac thread2.java
```

```
C:\java programs>java thread2
Seats booked successfully!!
Seats canceled successfully!!
```

```
C:\java programs>
```

Experiment 10- Collections

Ques1- Write a program for the following ArrayList (a) Read all elements from ArrayList by using Iterator, (b) Create a duplicate object of an ArrayList instance, (c) Reverse ArrayList content.

```
import java.util.*;
public class AL
{
    public static void main(String[] args)
    {
        ArrayList<Integer> l1 = new ArrayList<Integer>(); //CREATING ArrayList OBJECT
        l1.add(10);
        l1.add(20);
        l1.add(30);
        l1.add(40);
        l1.add(50);
        // (a) Read all elements from ArrayList by using Iterator
        Iterator i1 = l1.iterator();
        System.out.println("Original Arraylist-");
        while (i1.hasNext())
        {
            System.out.println(i1.next());
        }
        // (b) Create a duplicate object of an ArrayList instance
        ArrayList copy = new ArrayList();
        copy = (ArrayList)l1.clone();
        System.out.println("Duplicate ArrayList instance-"+copy);
        ArrayList l2=new ArrayList();
        // (c) Reverse ArrayList content
        System.out.print("Reversed Arraylist-");
        for(int i=l1.size()-1;i>=0;i--){
            l2.add(l1.get(i));
        }

        System.out.println(l2);
    }
}
```

```
C:\java programs> java AL
Original Arraylist-
10
20
30
40
50
Duplicate ArrayList instance-[10, 20, 30, 40, 50]
Reversed Arraylist-[50, 40, 30, 20, 10]
```

Ques2- Write a program for the following HashMap (a) Find whether the specified key exists or not, (b) Find whether the specified value exists or not, (c) Get all keys from the given HashMap, (d) Get all key-value pair as Entry objects.

```

import java.util.*;
public class HM
{
Scanner sc = new Scanner(System.in);
public static void main(String args[])
{
    HashMap<Integer, String> hm1 = new HashMap<Integer, String>();
hm1.put(1, "Vamika");
hm1.put(2, "Muskan");
hm1.put(3, "Riddhi");
hm1.put(4, "Khushi");
System.out.println("Hashmap-" + hm1);
//(a) Find whether the specified key exists or not
Scanner sc = new Scanner(System.in);
System.out.print("Enter key you want to search-");
int n = sc.nextInt();
if (hm1.containsKey(n)) {
System.out.println("Key is present");
}
else
{
System.out.println("Key is not present");
}
//(b) Find whether the specified value exists or not,
System.out.print("Enter value you want to search-");
String s = sc.next();
if (hm1.containsValue(s))
{
System.out.println("Value is present");
}
else
{
System.out.println("Value is not present");
}
//(c) Get all keys from the given HashMap
System.out.println("All keys from hm1 HashMap-");
for(Integer m:hm1.keySet())
{
System.out.println(m);
}
//(d) Get all key-value pair as Entry objects
System.out.println("All key-value pair as Entry objects-");
for (Map.Entry<Integer, String> entries : hm1.entrySet()) {
System.out.println(entries.getKey() + "-" + entries.getValue());
}
}
}

```

```
C:\java programs> java HM
Hashmap-{1=Vamika, 2=Muskan, 3=Riddhi, 4=Khushi}
Enter key you want to search- 1
Key is present
Enter value you want to search- Vamika
Value is present
All keys from hm1 HashMap-
1
2
3
4
All key-value pair as Entry objects-
1-Vamika
2-Muskan
3-Riddhi
4-Khushi

C:\java programs>
```

```
C:\java programs> java HM
Hashmap-{1=Vamika, 2=Muskan, 3=Riddhi, 4=Khushi}
Enter key you want to search- 7
Key is not present
Enter value you want to search- Nirmol
Value is not present
All keys from hm1 HashMap-
1
2
3
4
All key-value pair as Entry objects-
1-Vamika
2-Muskan
3-Riddhi
4-Khushi

C:\java programs>■
```

Ques3- Write a program for the following HashSet (a) Copy another collection object to the HashSet object, (b) Delete all entries at one call from HashSet, (c) Search user-defined objects from HashSet.

```
import java.util.*;
class HS
{
public static void main(String args[])
{
    HashSet<Integer> hs1=new HashSet<Integer>();
    hs1.add(111);
    hs1.add(222);
    hs1.add(333);
    System.out.println("Original Hashset-"+hs1);
    ArrayList<Integer> hs2=new ArrayList<Integer>();
    hs2.add(444);
    hs2.add(555);
    // (a) Copy another collection object to the HashSet object
    hs1.addAll(hs2);
    System.out.println("After Copying another collection object to the HashSet object-"+hs1);
    // (c) Search user-defined objects from HashSet
    Scanner sc=new Scanner(System.in);
    System.out.print("Enter element you want to search in Hashset-");
    int s=sc.nextInt();
    if(hs1.contains(s)){
        System.out.println("Element is present");
    }
    else{
        System.out.println("Element is not present");
    }
    // (b) Delete all entries at one call from HashSet
    hs1.clear();
    System.out.println("After deleting all entries at one call -"+hs1);
}}
```

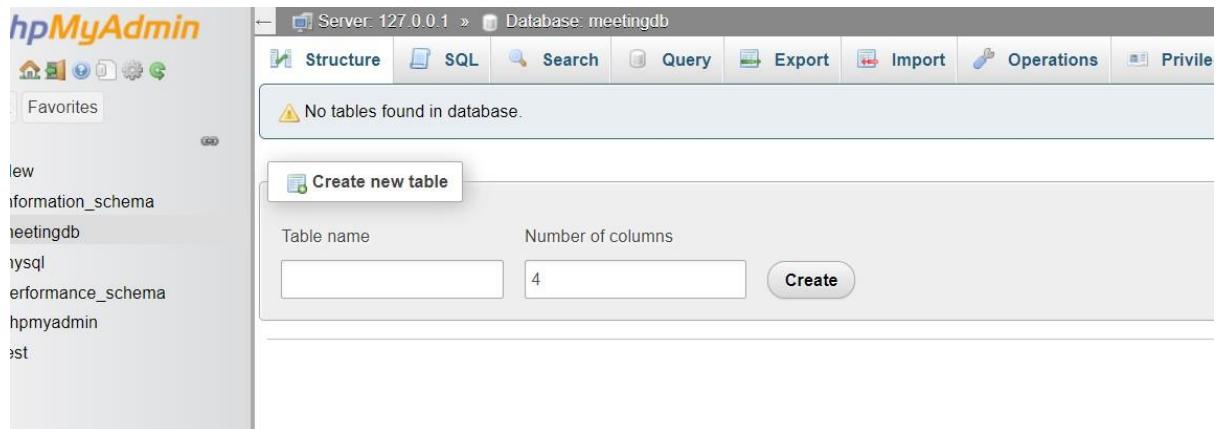
Command Prompt

```
C:\java programs>java HS
Original Hashset-[333, 222, 111]
After Copying another collection object to the HashSet object-[555, 444, 333, 222, 111]
Enter element you want to search in Hashset-555
Element is present
After deleting all entries at one call -[]

C:\java programs> java HS
Original Hashset-[333, 222, 111]
After Copying another collection object to the HashSet object-[555, 444, 333, 222, 111]
Enter element you want to search in Hashset-888
Element is not present
After deleting all entries at one call -[]

C:\java programs>
```

Experiment 11-JDBC



Q1. Create a table named "Meeting" having the columns (MeetingID, ScheduledTime, ScheduledDate, participantID, NameOfParticipant, participantDateBirth, Email, Mobile). "participantID" is set auto increment.

```
import java.sql.*;  
  
public class App  
{  
  
    public static void main(String[] args) throws Exception  
{  
  
        Class.forName("com.mysql.cj.jdbc.Driver");  
  
        Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/meetingDB", "root", "");  
  
        Statement statement = conn.createStatement();  
  
        statement.executeUpdate("CREATE TABLE Meeting (MeetingID int, ScheduledTime varchar(255),  
ScheduledDate Date, participantID int AUTO_INCREMENT , NameOfParticipant  
varchar(255), participantDateBirth Date, Email varchar(255), Mobile int, PRIMARY KEY(participantID))");  
  
    }  
  
}
```

```
C:\Users\Vamika> cd C:\java programs
C:\java programs> javac App.java
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	MeetingID	int(11)			Yes	NULL			Change Drop
2	ScheduledTime	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop
3	ScheduledDate	date			Yes	NULL			Change Drop
4	participantID	int(11)			No	None	AUTO_INCREMENT		Change Drop
5	NameOfParticipant	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop
6	participantDateBirth	date			Yes	NULL			Change Drop
7	Email	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop
8	Mobile	int(11)			Yes	NULL			Change Drop

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

```
SELECT * FROM `meeting`
```

Q2. Create a stored procedure that can be invoked to insert records to the table with proper data (5 records).

```
import java.sql.*;
import
java.util.Scanner;
class
Sample{

    public static void main(String[] args) {      try{
        //Connect to database
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/meetingdb", "root", "");      //Create Procedure
```

```

String query = "CREATE or REPLACE PROCEDURE `inrow`(MeetingID int, ScheduledTime
varchar(255), ScheduledDate varchar(10) , NameOfParticipant varchar(255), participantDateBirth
varchar(10), Email varchar(255), Mobile varchar(10)) " +
"BEGIN "+
" INSERT INTO meeting (MeetingID, ScheduledTime, ScheduledDate,
NameOfParticipant, participantDateBirth, Email, Mobile) VALUES(MeetingID,
ScheduledTime, ScheduledDate, NameOfParticipant, participantDateBirth, Email, Mobile); "+
"END";
Statement stmt = con.createStatement();      stmt.execute(query);
System.out.println("Procedure Created");
// Input and insert
Scanner obj = new Scanner(System.in);
int mid;
String scht,schd,np,pd,email,mob;
CallableStatement stmtm = con.prepareCall("{call inrow(?,?,?,?,?,?)}");
for (int i = 0; i < 5; i++) {
System.out.println("MeetingId");
mid = obj.nextInt();
stmtm.setInt(1, mid);
System.out.println("ScheduledTime");
scht = obj.next();
stmtm.setString(2, scht);
System.out.println("ScheduledDate");
schd = obj.next();
stmtm.setString(3, schd);
System.out.println("NameofPaticipant");
np = obj.next();
stmtm.setString(4, np);
System.out.println("ParticipantDateBirth");
pd = obj.next();
stmtm.setString(5, pd);
System.out.println("Email");
email = obj.next();
stmtm.setString(6, email);
}

```

```

        System.out.println("Mobile");
        mob = obj.next();
        stmtm.setString(7, mob);
        stmtm.execute();
    }
    obj.close();
    con.close();

}catch(Exception e){
    System.out.println(e);
}
}
}
}

```

```

Procedure Created
MeetingId
14
ScheduledTime
14:10
ScheduledDate
14-10-2022
NameofParticipant
a
ParticipantDateBirth
10-9-1998
Email
mail@mail
Mobile
9898989898
MeetingId
14
ScheduledTime
14:10
ScheduledDate
14-10-2022
NameofParticipant
b
ParticipantDateBirth
12-01-1989
Email
em@mail
Mobile

```

phpMyAdmin

Server: 127.0.0.1 » Database: meetingdb » Table: meeting

Browse Structure SQL Search Insert Export Import Privileges Operations More

T * FROM `meeting`

Filtering [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Actions

	MeetingID	ScheduledTime	ScheduledDate	participantID	NameOfParticipant	participantDateBirth	Email	Mobile
Edit	14	14:10	14-10-2022	1	a	10-9-1998	mail@mail	989898989898
Edit	14	14:10	14-10-2022	2	b	12-01-1989	em@mail	999999999999
Edit	14	14:10	14-10-2022	3	c	05-05-1999	tw@mail	888888888888
Edit	15	20:12:2022	20-12-2022	4	t	12-12-1995	m@mail	9765432180
Edit	18	16:00	15-01-2023	5	nt	04-07-1994	ma@mail	9087654321

Check all With selected:

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Routines

Check all

	Name	Type	Returns
<input type="checkbox"/>	inrow	PROCEDURE	<input type="button" value="Edit"/> <input type="button" value="Execute"/> <input type="button" value="Export"/> <input type="button" value="Drop"/>

Edit

Routine name:

Type: PROCEDURE

Direction	Name	Type
IN	MeetingID	INT
IN	ScheduledTime	VARCHAR
IN	ScheduledDate	VARCHAR
IN	NameOfParticipant	VARCHAR
IN	participantDateBirth	VARCHAR
IN	Email	VARCHAR
IN	Mobile	VARCHAR

Parameters

Q3. Get the "MeetingID" from the user. Display the names of all the participants for the meeting with the given MeetingID on the console, preceded by the message, “Names of participants in meeting 1144”.

```

import java.sql.*;
import
java.util.Scanner;
public class App {
    public static void main(String[] args) throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/meetingDB", "root", "");
        Statement statement = conn.createStatement();
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the meeting ID: ");
        int meetingID = scanner.nextInt();
        scanner.close();
        String query = String.format("SELECT NameOfParticipant FROM Meeting WHERE meetingID = %d;", meetingID);
        ResultSet resSet = statement.executeQuery(query);
        System.out.println(String.format("Names of participants in meeting %d: ", meetingID));
        while (resSet.next()) {
            System.out.println(resSet.getString("NameOfParticipant"));
        }
    }
}

```

	MeetingID	ScheduledTime	ScheduledDate	participantID	NameOfParticipant	participantDateBirth	Email
<input type="checkbox"/>   	1105	10:00	2022-12-04	1	John	1998-10-10	john@gmai
<input type="checkbox"/>   	1105	10:00	2022-12-04	2	Mary	2000-04-20	mary@gma
<input type="checkbox"/>   	1105	10:00	2022-12-04	3	Alex	1994-04-01	Alex@gmai
<input type="checkbox"/>   	1105	10:00	2022-12-04	4	Anna	2000-08-01	Anna@gma
<input type="checkbox"/>   	1105	10:00	2022-12-04	5	Harry	1996-02-10	Harry@gma
<input type="checkbox"/>   	1105	10:00	2022-12-04	6	Kevin	1992-12-12	Kevin@gma

Q4. Insert a new participant using only the participant's name. Do not provide the "participantID". Determine the ID that was assigned to the new participant and display it on the console (the statement that displays the ID should display the name of the new participant as well).

```
import java.sql.*;
import java.util.Scanner;
import javax.management.Query; class
Sample{

    public static void main(String[] args) {      try{
        //Connect to database
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/meetingdb", "root", "");
        Scanner obj = new Scanner(System.in);
        System.out.println("Name");
        String name = obj.next();
        String query1 = "INSERT INTO meeting (NameOfParticipant) VALUES(?)";
query2;
        PreparedStatement stmt = con.prepareStatement(query1);
        stmt.setString(1, name);      stmt.execute();
        System.out.println("Query Executed");
        query2 = "SELECT * from meeting WHERE participantId =(SELECT
MAX(participantID) from meeting);";
        Statement sttm = con.createStatement();      ResultSet rs
= sttm.executeQuery(query2);      while(rs.next()){
            System.out.println("Name: "+rs.getString(5)+" participantId: "+rs.getInt(4));
        }
        obj.close();
        con.close();
    }catch(Exception e){
        System.out.println(e);
    }
}
```

Output:

```
Name
participant1
Query Executed
Name: participant1 participantId: 8

C:\Users\laksh\OneDrive\Desktop\lakshit\college>
```

The screenshot shows the phpMyAdmin interface for a database named 'meetingdb'. On the left, the database structure is visible with a tree view showing 'information_schema', 'meetingdb' (selected), 'mysql', 'performance_schema', and 'test'. Under 'Tables', there is a single entry for 'meeting'. The main right panel displays the contents of the 'meeting' table. The table has columns: MeetingID, ScheduledTime, ScheduledDate, participantID, NameOfParticipant, participantDateBirth, Email, and Mobile. There are 7 rows of data. Row 1: MeetingID 14, ScheduledTime 14:10, ScheduledDate 14-10-2022, participantID 1, NameOfParticipant a, participantDateBirth 10-9-1998, Email mail@mail, Mobile 9898989898. Row 2: MeetingID 14, ScheduledTime 14:10, ScheduledDate 14-10-2022, participantID 2, NameOfParticipant b, participantDateBirth 12-01-1989, Email em@mail, Mobile 9999999999. Row 3: MeetingID 14, ScheduledTime 14:10, ScheduledDate 14-10-2022, participantID 3, NameOfParticipant c, participantDateBirth 05-05-1999, Email tw@mail, Mobile 8888888888. Row 4: MeetingID 15, ScheduledTime 20:12:2022, ScheduledDate 20-12-2022, participantID 4, NameOfParticipant t, participantDateBirth 12-12-1995, Email m@mail, Mobile 9765432180. Row 5: MeetingID 18, ScheduledTime 16:00, ScheduledDate 15-01-2023, participantID 5, NameOfParticipant nt, participantDateBirth 04-07-1994, Email ma@mail, Mobile 9087654321. Row 6: MeetingID NULL, ScheduledTime NULL, ScheduledDate NULL, participantID 8, NameOfParticipant participant1, participantDateBirth NULL, Email NULL, Mobile NULL. Row 7: MeetingID NULL, ScheduledTime NULL, ScheduledDate NULL, participantID 8, NameOfParticipant participant1, participantDateBirth NULL, Email NULL, Mobile NULL.

Q5. Count the number of people participating in the meeting with ID 1105. Display a message on the console that gives both the "MeetingID" and the number of participants.

```
import java.sql.*; import
java.util.Scanner; public
class App {

    public static void main(String[] args) throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/meetingDB", "root", "");
        Statement statement = conn.createStatement();      int meetingID = 1105;

        String query = String.format("SELECT NameOfParticipant FROM Meeting WHERE meetingID = %d;", meetingID);

        ResultSet resSet = statement.executeQuery(query);
        int count = 0;      while (resSet.next()) {           count++;
    }

    System.out.println(String.format("Meeting ID: 1105 || Participants: %d", count));
}
```

```
PS C:\projects\java_project_domestic\App\src> javac App.java
PS C:\projects\java_project_domestic\App\src> java App
Meeting ID: 1105 || Participants: 6
```

	<input type="button" value="←"/> <input type="button" value="→"/>	MeetingID	ScheduledTime	ScheduledDate	participantID	NameOfParticipant	participantDateBirth	Email
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1105	10:00	2022-12-04	1	John	1998-10-10	john@gmai
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1105	10:00	2022-12-04	2	Mary	2000-04-20	mary@gma
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1105	10:00	2022-12-04	3	Alex	1994-04-01	Alex@gmai
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1105	10:00	2022-12-04	4	Anna	2000-08-01	Anna@gma
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1105	10:00	2022-12-04	5	Harry	1996-02-10	Harry@gma
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1105	10:00	2022-12-04	6	Kevin	1992-12-12	Kevin@gma

Q6. Determine the names of all who participate in meetings on Tuesdays.

Display the names, preceded by the message, “Participants attending Tuesday meetings”.

```
import java.sql.*;
class Sample{
    public static void main(String[] args) {
        try{
            //Connect to database
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con =
                DriverManager.getConnection("jdbc:mysql://localhost:3306/meetingdb", "root", "");
            Statement stmt =
                con.createStatement();
            String query = "Select NameOfParticipant from meeting WHERE
WEEKDAY(ScheduledDate) = 1";
            ResultSet rs = stmt.executeQuery(query);
            System.out.print("Participants attending Tuesday's meeting ");
            while(rs.next()){
                System.out.print(rs.getString(1)+", ");
            }
            con.close();
        }catch(Exception e){
            System.out.println(e);
        }
    }
}
```

```
Participants attending Tuesday's meeting ParticipantA, ParticipantC, ParticipantE,
C:\Users\laksh\OneDrive\Desktop\lakshit\college>
```

The screenshot shows the phpMyAdmin interface for a database named 'meetingdb'. The 'Tables' section is selected, and the 'meeting' table is displayed. The table has the following columns: MeetingID, ScheduledTime, ScheduledDate, participantID, NameOfParticipant, participantDateBirth, Email, and Mobile. There are five rows of data:

	MeetingID	ScheduledTime	ScheduledDate	participantID	NameOfParticipant	participantDateBirth	Email	Mobile
1	10	2022-12-04	2022-12-06	9	ParticipantA	2012-02-08	mail@mail	9890102309
2	10	2022-12-06	2022-12-06	10	ParticipantC	1996-02-08	ABC@mail	9891202345
3	14	2022-12-11	2022-12-21	11	ParticipantB	1995-12-09	ko@mail	9654321908
4	20	2022-12-02	2022-12-16	12	ParticipantD	1996-12-30	hyt@mail	9818987609
5	34	2022-12-20	2022-12-20	13	ParticipantE	1996-12-14	po@mail	9812345670

Q7. Create a function to get the "participantDateBirth", find the participant's age, and print it.

```
import java.sql.*;

public class App {

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

        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/meetingDB", "root", "");

        Statement statement = conn.createStatement();

        String query = "SELECT NameOfParticipant FROM Meeting WHERE WEEKDAY(ScheduledDate) = 1;";

        ResultSet resSet = statement.executeQuery(query);

        boolean remaining = resSet.next();

        if (!remaining){

            System.out.println("No participants attending Tuesday meeting");

        }

    else {
```

```

System.out.println("Participants attending Tuesday Meeting:");

while (remaining){

    System.out.println(resSet.getString("NameOfParticipant"));

    remaining = resSet.next();

}

}

}

```

	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	MeetingID	ScheduledTime	ScheduledDate	participantID	NameOfParticipant	participantDateBirth	Email
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1105	10:00	2022-12-06	1	John	1998-10-10	john@gmai
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1105	10:00	2022-12-06	2	Mary	2000-04-20	mary@gma
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1105	10:00	2022-12-06	3	Alex	1994-04-01	Alex@gmai
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1105	10:00	2022-12-06	4	Anna	2000-08-01	Anna@gma
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1105	10:00	2022-12-06	5	Harry	1996-02-10	Harry@gma
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1105	10:00	2022-12-06	6	Kevin	1992-12-12	Kevin@gma

Experiment 12- Servlets And JSP

Q1. Write a Servlet page to create a simple calculator.

index.html

```
<html>
<head>

<title>Calculator App</title>

</head>

<body>

<form action="CalculatorServlet" method="post" >

Enter First Number <input type="text" name="txtN1" ><br> Enter

Second Number <input type="text" name="txtN2" ><br> Select an

Operation

<input type="radio" name="opr" value="+">ADDITION <input type="radio" name="opr"
value="* ">MULTIPLY <input type="radio" name="opr" value="/">DIVIDE

<input type="radio" name="opr" value="-"> Substraction

<br> <input type="reset">

<input type="submit" value="Calculate" >

</form>

</body>

</html>
```

CalculatorServlet.java

```
package mypack;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class CalculatorServlet extends HttpServlet
{
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
```

```

response.setContentType("text/html;charset=UTF-8);

PrintWriter out = response.getWriter();

    out.println("<html><head><title>Servlet
CalculatorServlet</title></head><body>");

        double n1 =
Double.parseDouble(request.getParameter("txtN1"));

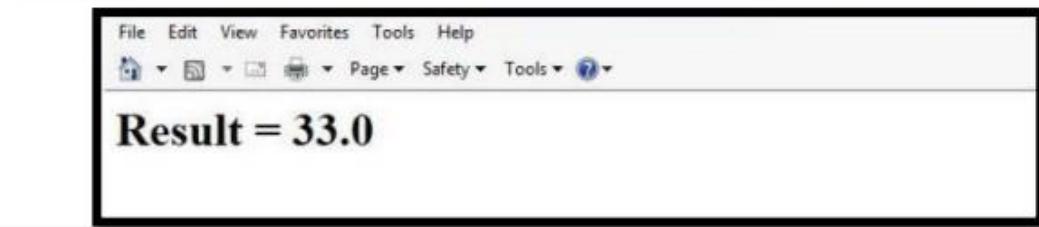
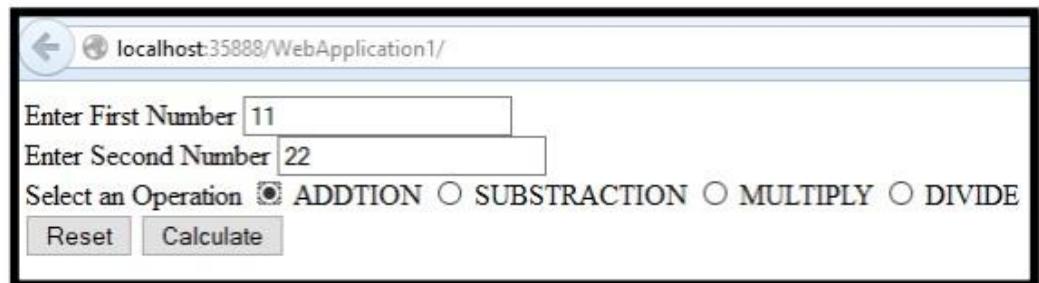
double n2 = Double.parseDouble(request.getParameter("txtN2")); double result =0;

String opr=request.getParameter("opr"); if(opr.equals("+")) result=n1+n2;
if(opr.equals("-")) result=n1-n2; if(opr.equals("*")) result=n1*n2;
if(opr.equals("/")) result=n1/n2;

out.println("<h1> Result = "+result);
out.println("</body></html>");

}
}

```



Q2. Write a Servlet page that includes two other Servlet pages.

Index.html

```

<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">    </head>
    <body>
        <form action="servlet1" method="get">      number1: <input type="text" name="no1">
<br> <br>      number2: <input type="text" name="no2"> <br> <br>

```

```

<input type="submit" name="btnadd" value="Add">
</form>
</body>
</html>

```

Add.java

```

package new1;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpServletResponse;
public class add extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            int a=Integer.parseInt(request.getParameter("no1"));
            int b=Integer.parseInt(request.getParameter("no2"));
            int c=(a+b);
            if(c%2==0){
                out.println("Addition is: "+(a+b));
            }
            else{
                RequestDispatcher rd=request.getRequestDispatcher("servlet2");
                rd.forward(request, response);
            }
        }
    }
}

```

NewServlet.java

```

package new1;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;

```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpServletResponse;

public class NewServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        out.println("Addition is odd");
    }
}

```

Q3. Write a JSP page to access the data of a student from the student table.

welcome_to_database_query.jsp

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<%@ page import="java.sql.*" %>
<%@ page import="java.io.*" %>

<html>
    <head>
        <title>display data from the table using jsp</title>
    </head>
    <body>
        <TABLE style="background-color: #ffffcc;">
            <TR>
                <TD align="center">
                    <h2>To display all the data from the table click here...</h2></TD>
                </TR>
                <TR>
                    <TD align="center"><A
HREF="ConnectJspToMysql.jsp">
                        <font size="4" color="blue">show data from
                            table</font></A></TD>

```

```

        </TR>
    </TABLE>
</body>
</html>

```

ConnectJspToMysql.jsp

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<%@ page import="java.sql.*" %>
<%@ page import="java.io.*" %>

<html>
<head>
    <title>display data from the table using jsp</title>
</head>
<body>
    <h2>Data from the table 'stu_info' of database 'student'</h2>
    <%
    try {
        String connectionURL = "jdbc:mysql://localhost:3306/student";
        Connection connection = null;
        Statement statement = null
        ResultSet rs = null;
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        connection = DriverManager.getConnection(connectionURL, "root", "root");
        statement = connection.createStatement();
        String QueryString = "SELECT * from stu_info";      rs =
        statement.executeQuery(QueryString);
    %>
    <TABLE cellpadding="15" border="1" style="background-color:
#ffffcc;">    <%
        while (rs.next()) {

```

```

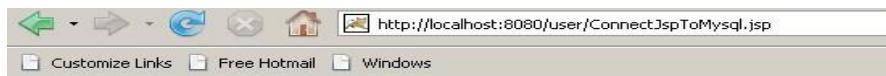
%>

<TR>
    <TD><%=rs.getInt(1)%></TD>
    <TD><%=rs.getString(2)%></TD>
    <TD><%=rs.getString(3)%></TD>
    <TD><%=rs.getString(4)%></TD>
</TR>
<%  }  %>
<%
    rs.close();  statement.close();
connection.close(); } catch (Exception
ex) {
    %>
</font>
<font size="+3" color="red"></b>
<%
    out.println("Unable to connect to database.");
}
%>
</TABLE><TABLE>
<TR>
    <TD><FORM ACTION="welcome_to_database_query.jsp" method="get" >
        <button type="submit"><- back</button></TD>
</TR>
</TABLE>
</font>
</body>
</html>

```

To display all the data from the table click here...

[show data from table](#)



Data from the table 'stu_info' of database 'student'

101	Amit	Mubarikpur	9811896358
70	mahendra	delhi	9990254913
80	Ravi	delhi	9911635841

Q4. Write a JSP Login page to get the username and password from the user. Display a welcome page on successful login or display the wrong authentication page.

login.jsp

```
<html>
<head>
<title>Login Form</title>
</head>
<body>
<h3> Login here </h3>
<form action="user_login" method="post">
<table style="width: 20%">
<tr>
<td>UserName</td>
<td><input type="text" name="username" /></td>
</tr>
<tr>
<td>Password</td>
<td><input type="password" name="password" /></td>
</tr>
</table>
<input type="submit" value="Login" /></form>
</body> </html>
```

user_login.java(servlet)

```
package demotest;
```

```

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpServletResponse;
; public class user_login extends
HttpServlet { public user_login() {
super();}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
String username = request.getParameter("username"); String password =
request.getParameter("password"); if(username.isEmpty() || password.isEmpty() )

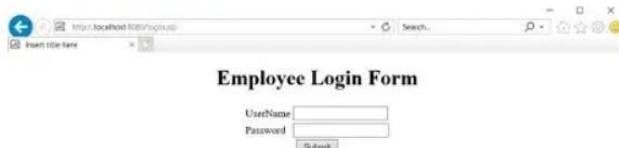
{
RequestDispatcher requ = request.getRequestDispatcher("login.jsp"); requ.include(request, response);

} else{

RequestDispatcher requ =
request.getRequestDispatcher("login_2.jsp"); requ.forward(request, response);

}}}

```



Thank You